ĐẠI HỌC QUỐC GIA TP.HCM

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

MÔN HỌC: AN TOÀN MẠNG

BÁO CÁO ĐỒ ÁN

Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing

Phát hiện DGA Botnet mạnh mẽ: Kết hợp XAI và OSINT để chia sẻ thông tin về mối đe dọa mạng

Giảng viên hướng dẫn: Nghi Hoàng Khoa

NT140.O11.ATCL

Sinh viên thực hiện :

21522492 - Ngô Minh Quân
21522312 - Phùng Đức Lương
21522483 - Chu Nguyễn Hoàng Phương

TP.HCM,Ngày 14 tháng 1 năm 2024

MỤC LỤC

DANH MỤC CÁC KÝ HIỆU, CÁC CHỮ VIẾT TẮT

CharLength	domain name character length
CTI	cyber threat intelligence
DGA	domain generation algorithm
IoC	indicators of compromise
IRad	information radius
Min-RE-Botnets	minimum of relative entropy botnets
OSINT	open-source intelligence
RE	relative entropy
RE-Alexa	the RE of a domain name to Alexa
ReputationAlex	Alexa reputation score
TreeNewFeature	a new feature generated by decision tree
XAI	explainable artificial intelligence

DANH MỤC CÁC HÌNH VẼ

DANH MỤC CÁC BẢNG BIỂU

LỜI MỞ ĐẦU - TÓM TẮT ĐỀ TÀI

Botnet là công cụ mạnh mẽ được tội phạm mạng sử dụng để thực hiện nhiều hoạt động độc hại khác nhau, chẳng hạn như tấn công từ chối dịch vụ phân tán (DDoS), chiến dịch thư rác và đánh cắp dữ liệu. Phát hiện và giảm thiểu botnet là một nhiệm vụ quan trọng đối vớicác chuyên gia an ninh mạng để bảo vệ mạng và hệ thống khỏi tác hại của chúng. Một kỹ thuật thường được các botnet sử dụng là Thuật toán tạo tên miền (DGA), thuật toán này tạo ra một số lượng lớn tên miền để thiết lập các kênh truyền thông chỉ huy và kiểm soát (C&C).

Để phát hiện và chống lại các botnet sử dụng DGA một cách hiệu quả, cần có một cách tiếp cận mạnh mẽ, kết hợp các công nghệ tiên tiến như Trí tuệ nhân tạo có thể giải thích (XAI) và Thông tin nguồn mở (OSINT) với Thông tin mối đe dọa mạng (CTI). Sự tích hợp này cho phép chúng ta tận dụng các phân tích nâng cao, học máy và thông tin thời gian thực để nâng cao khả năng phát hiện mạng botnet của chúng ta. Trí tuệ nhân tạo có thể giải thích (XAI) đóng một vai trò quan trọng trong việc tìm hiểu hoạt động bên trong của các mô hình học máy. Bằng cách sử dụng kỹ thuật XAI, chúng ta có thể diễn giải các quyết định và dự đoán do hệ thống AI đưa ra, khiến chúng trở nên minh bạch và đáng tin cậy hơn.

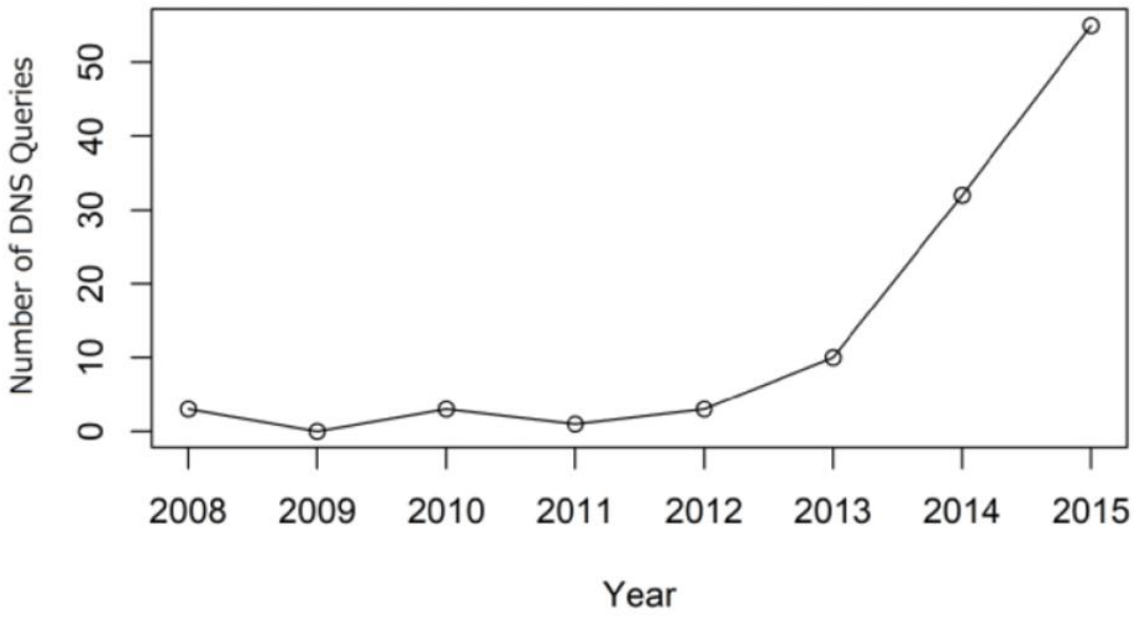
Thông tin nguồn mở (OSINT) liên quan đến việc thu thập và phân tích dữ liệu có sẵn công khai từ nhiều nguồn khác nhau, chẳng hạn như phương tiện truyền thông xã hội, trang web và diễn đàn. OSINT cung cấp thông tin có giá trị về các hoạt động botnet đã biết, các chỉ số xâm phạm (IoC) cũng như các chiến thuật, kỹ thuật và thủ tục (TTPs) được tội phạm mạng sử dụng. Thông tin về mối đe dọa mạng (CTI) đề cập đến kiến thức và hiểu biết sâu sắc thu được từ việc phân tích dữ liệu liên quan đến các mối đe dọa và đối thủ trên mạng. Nó bao gồm thông tin về hồ sơ kẻ tấn công, động cơ và phương pháp của chúng. Bằng cách tận dụng CTI, các chuyên gia an ninh mạng có thể chủ động dự đoán và ứng phó với các hoạt động của botnet, nâng cao khả năng phát hiện và giảm thiểu. Việc kết hợp XAI, OSINT và CTI cho phép tiếp cận toàn diện và chủ động để phát hiện botnet. Bằng cách kết hợp khả năng diễn giải của các kỹ thuật XAI, thông tin thời gian thực từ các nguồn OSINT và những hiểu biết sâu sắc về chiến lược từ CTI, chúng ta có thể xây dựng các cơ chế phòng thủ hiệu quả và linh hoạt hơn trước các mối đe dọa từ mạng botnet.

CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN

1.1. Đặt vấn đề

Botnet là mối đe dọa gây ra nhiều vụ tấn công mạng nghiêm trọng. Nó sử dụng máy chủ ra lệnh và điều khiển (C&C). Botnet sử dụng thuật toán sinh tên miền động (DGA) là kiểu botnet khó phát hiện nhất bởi chúng sử dụng nhiều tên miền máy chủ C&C được tạo ra theo thuật toán. Bot quét máy chủ C&C thông qua giao thức tên miền (DNS) để che giấu thông điệp liên lạc. Do đó, hệ thống bảo vệ an ninh gặp khó khăn trong phát hiện .

Chúng tôi đã phân tích tập dữ liệu truy vấn máy chủ DNS 12 năm từ máy chủ cache đầy đủ của trường đại học chúng tôi. Đáng ngạc nhiên, 107 truy vấn DNS (tên miền độc hại) đã được phát hiện là lưu lượng botnet. Kết quả này được thu được bằng danh sách chặn chứa 803.333 tên miền độc hại; tuy nhiên, do đặc điểm của thuật toán sinh tên miền DGA, chỉ dựa vào danh sách chặn có thể không đủ. Điều này cho thấy cần phải áp dụng mô hình học máy/AI cho hệ thống CTI.



Hình 1. Những thay đổi trong lưu lượng DGA của botnet độc hại

1.2. Các công trình nghiên cứu liên quan

Mặc dù có nhiều cách tiếp cận để bảo vệ hệ thống tên miền chống lại botnet (che giấu giao tiếp với máy chủ C&C), cách tiếp cận bảo mật tên miền truyền thống dựa trên danh sách chặn gặp vấn đề vì không có danh sách nào hoàn toàn đáng tin cậy.

Trong một ấn phẩm gần đây về phát hiện botnet DGA, Hoang và Vu đề xuất mô hình rừng ngẫu nhiên cải tiến bằng cách tính toán 24 đặc trưng thống kê như phân phối tần suất n-gram ký tự của tên miền, giá trị entropy, ký tự đầu tiên có phải số hay

không, và nhiều tính toán thống kê khác. Kết quả thử nghiệm trên tập dữ liệu 39 gia đình DGA cho thấy hiệu suất cao hơn các công trình trước. Tuy nhiên, tính toán 24 đặc trưng có thể làm tăng độ phức tạp tính toán. Do đó, chúng tôi đề xuất chỉ sử dụng 7 đặc trưng: entropy, entropy tương đối Alexa, giá trị tối thiểu của entropy botnets, bán kính thông tin (Irad), độ dài ký tự, đặc trưng mới tạo bằng thuật toán cây quyết định, và điểm uy tín của tên miền. Ngoài ra, chúng tôi thử nghiệm với phạm vi rộng hơn các tập dữ liệu DGA (tổng cộng 55 gia đình DGA).

Ngoài phương pháp dựa trên đặc trưng thống kê, phương pháp phân loại dựa trên ký tự, tận dụng bối cảnh ký tự của tên miền, cũng có thể sử dụng cho phân loại DGA. Biên soạn các mô hình học sâu dựa trên ký tự với các kiến trúc khác nhau cho phân loại DGA. Để đánh giá công trình của chúng tôi, chúng tôi triển khai bốn mô hình học sâu: Endgame, CMU, NYU, và MIT.

Các cuộc tấn công DGA tiên tiến hiện nay có thể thấy trong ChatBot, DeepDGA và MaskDGA , nơi các tác giả áp dụng các phương pháp tinh vi như tấn công đối nghịch ML, mẫu đối nghịch để sinh ra tên miền tránh bị phân loại DGA. Sidi. Chứng minh rằng cuộc tấn công MaskDGA làm giảm hiệu suất bộ phân loại DGA, tránh bị phát hiện. Peck. cho thấy hiệu quả của cuộc tấn công CharBot, giảm tỉ lệ phát hiện của bộ phân loại xuống thấp như 1,69%; ngay cả việc huấn luyện lại bộ phân loại cũng không phải là chiến lược phòng thủ khả thi.

Chúng tôi áp dụng ba cuộc tấn công DGA (CharBot, DeepDGA và MaskDGA) để kiểm tra hiệu suất của mô hình phát hiện botnet DGA của chúng tôi trong đối phó với các cuộc tấn công đối nghịch hiện đại.

Bảng 1: Những phương pháp XAI để cải thiện độ tin cậy khi chia sẻ CTI

Taxonomy	ANCH-OR	LIME	SH-AP	Counter-factual	XAI-OSINT
Explanation by simplification:					
Rule-based learner	√	√	-	-	-
Decision tree	-	-	-	-	-
Feature relevance explanation:					
Influence function	-	-	-	-	-
Sensitivity	-	-	-	-	-
Game Theory inspired	-	-	√	-	-
Interaction based	-	-	-	-	-
Local explanation:					
Rule-based learner	√	√	-	-	-
Decision tree	-	-	-	-	-
Visual explanation:					
Conditional/dependence/ Shapley plots	-	-	√	-	-
Sensitivity/saliency	-	-	-	-	-
Explanation by example:					
Counterfactual explanation	-	-	-	√	-
Our proposed approach:					
Second opinion	-	-	-	-	√

CHƯƠNG 2. KIẾN THỨC NỀN TẢNG

Chương này trình bày sơ lược cơ sở lý thuyết của nghiên cứu bao gồm: Botnet và DGA, XAI, OSINT, CTI, Machine Learning.

2.1 Botnet và DGA

Botnet là một hệ thống máy tính hoặc thiết bị kết nối với internet và được kiểm soát từ xa bởi một người hay tổ chức tấn công mà không sự cho phép của chủ sở hữu. Những máy tính này, thường được cài đặt phần mềm độc hại mà người điều khiển có thể sử dụng để thực hiện các hoạt động độc hại như tấn công phủ định dịch vụ (DDoS), lừa đảo thông tin cá nhân, hoặc thậm chí làm nhiệm vụ như làm máy chủ Command and Control (C2) cho các cuộc tấn công phức tạp hơn.

Domain Generation Algorithm (DGA) là một phần quan trọng của botnet, giúp tạo ra các tên miền được sử dụng để kết nối các máy tính trong botnet với máy chủ điều khiển. Thay vì sử dụng một tên miền cố định, DGA tạo ra các tên miền mới động, thường dựa trên một thuật toán phức tạp và các thông số như thời gian hoặc thông tin đặc biệt về hệ thống, để làm cho việc phát hiện và chặn trở ngại trở ngại của hệ thống bảo mật trở nên khó khăn hơn. Cách tiếp cận này giúp botnet duy trì tính ổn định và khả năng ẩn náu, làm tăng khả năng thành công của các hoạt động tấn công mà không bị phát hiện sớm.

2.2 XAI

XAI, hay Explainable Artificial Intelligence, là một lĩnh vực quan trọng trong nghiên cứu và phát triển trí tuệ nhân tạo (AI). Mục tiêu chính của XAI là làm cho quá trình ra quyết định của các mô hình máy học trở nên minh bạch và dễ giải thích đối với con người. Trong khi các mô hình máy học truyền thống thường xuyên được coi là "hộp đen" - nghĩa là quá trình ra quyết định của chúng khá khó hiểu và không rõ ràng - XAI mục tiêu giải quyết vấn đề này bằng cách tạo ra mô hình giải thích được.

Mô hình giải thích được không chỉ cung cấp kết quả dự đoán mà còn giải thích cách mà mô hình đã đưa ra quyết định đó. Điều này giúp người dùng, bao gồm cả những người không chuyên về lĩnh vực máy học, hiểu rõ hơn về cơ sở lý do của mô hình, từ đó tăng cường sự tin cậy và chấp nhận của nó trong các ứng dụng quan trọng như y tế, tài chính, hay an ninh mạng.

Trong bối cảnh chủ đề "Robust Botnet DGA Detection," việc tích hợp XAI vào các mô hình phát hiện botnet có thể cung cấp cái nhìn sâu sắc hơn về cách mà hệ thống đưa ra quyết định về việc xác định các hoạt động đáng ngờ và nó có thể giúp người nghiên cứu hay chuyên gia an ninh mạng hiểu rõ hơn về cơ sở lý do của các dự đoán, từ đó nâng cao khả năng phát hiện và đối phó với botnet.

2.3 OSINT

Open Source Intelligence (OSINT) đóng vai trò quan trọng trong lĩnh vực an ninh mạng và tình báo, đặc biệt là trong ngữ cảnh ngày nay khi môi trường kỹ thuật số ngày càng phức tạp và đa dạng. OSINT tập trung vào việc thu thập, phân tích và tận dụng thông tin từ các nguồn mở trực tuyến và các nguồn thông tin công cộng khác, mà không yêu cầu sự tương tác trực tiếp với các đối tượng nghiên cứu. Điều này bao gồm truy cập các trang web, diễn đàn, mạng xã hội, bản tin, và một loạt các nguồn khác có sẵn trên internet.

Mục tiêu chính của OSINT là cung cấp cái nhìn toàn diện về môi trường an ninh mạng, giúp tổ chức hiểu rõ hơn về mối đe dọa tiềm ẩn và chiến lược tấn công đang được triển khai. Bằng cách thu thập thông tin từ các nguồn mở, OSINT có thể giúp xác định địa chỉ IP, tên miền, thông tin về tổ chức, người sử dụng, và các thông tin

khác có thể hỗ trợ trong việc đánh giá rủi ro và phòng ngự trước các mối đe dọa an ninh mạng.

2.4 CTI

Cyber Threat Intelligence (CTI) là một lĩnh vực quan trọng trong lĩnh vực an ninh mạng, chuyên nghiên cứu và ứng dụng các phương pháp thu thập, phân tích, và chia sẻ thông tin liên quan đến các mối đe dọa an ninh mạng. Mục tiêu của CTI là cung cấp cái nhìn tổng thể và chi tiết về các mối đe dọa tiềm ẩn, chiến lược tấn công, và các biểu hiện của hoạt động độc hại trực tuyến. Bằng cách này, CTI chơi một vai trò quan trọng trong việc cung cấp thông tin quan trọng cho tổ chức, giúp họ nâng cao khả năng phát hiện và ứng phó với các cuộc tấn công mạng ngày càng tinh vi và phức tạp.

Trong ngữ cảnh của chủ đề "Robust Botnet DGA Detection," CTI không chỉ giúp xác định các đặc điểm của botnet mà còn cung cấp thông tin chi tiết về các chiến lược tấn công mà chúng sử dụng, bao gồm cả việc sử dụng Domain Generation Algorithms (DGA). CTI có thể bao gồm các bản đánh giá rủi ro, thông tin về đối tượng hay nhóm tấn công, và các chiến thuật phòng ngự được áp dụng để chống lại những mối đe dọa này.

Tích hợp CTI vào quá trình phân tích và phát hiện botnet giúp nâng cao chính xác và độ nhạy của hệ thống an ninh mạng. Sự hiểu biết sâu sắc về mối đe dọa từ CTI cung cấp cơ sở cho việc phát triển và cải tiến các giải pháp bảo mật, đồng thời giúp tổ chức chuẩn bị sẵn sàng đối mặt với các mối đe dọa tiềm ẩn trong tương lai.

2.5. Mô hình học máy (Machine Learning)

Học máy hay máy học (Machine Learning) là một loại trí tuệ nhân tạo (AI) cho phép các ứng dụng phần mềm trở nên chính xác hơn trong việc dự đoán kết quả mà không cần được lập trình rõ ràng để làm như vậy. Các thuật toán học máy sử dụng dữ liệu lịch sử làm đầu vào để dự đoán các giá trị đầu ra mới.

Một số ứng dụng phổ biến của học máy bao gồm hệ thống đề xuất (Recommend System), hệ thống phát hiện bất thường (Anomaly Detection), phát hiện xâm nhập (IDS), phần mềm độc hại (Malwares), lọc thư rác (Spam email) ... Học máy cổ điển thường được phân loại theo cách một thuật toán học để trở nên chính xác hơn trong các dự đoán của nó. Có bốn cách tiếp cận cơ bản: học có giám sát, học không giám sát, học bán giám sát và học tăng cường. Chúng tôi đã sử dụng nhiều thuật toán ML được giám sát để so sánh kết quả nhằm chọn ra thuật toán có độ chính xác tốt nhất. Ở đây, năm thuật toán (naive Bayes, logistic regression, extra tree, random forest, and ensemble learning) đã được tính toán bằng Scikit-Learn

- Naive Bayes: Naive Bayes là một thuật toán phân loại dựa trên nguyên tắc của định lý Bayes. Nó giả định rằng các đặc trưng đầu vào độc lập với nhau và sử dụng xác suất để dự đoán lớp của một mẫu mới.

- Logistic Regression: Logistic Regression là một thuật toán phân loại dựa trên mô hình hồi quy logistic. Nó sử dụng hàm logistic để ước lượng xác suất của một mẫu thuộc về một lớp cụ thể và thực hiện việc phân loại dựa trên ngưỡng xác suất.
- Extra Trees: Extra Trees (Extreme Randomized Trees) là một biến thể của thuật toán Random Forest. Nó sử dụng một tập hợp lớn các cây quyết định ngẫu nhiên để thực hiện phân loại hoặc hồi quy. Extra Trees giảm thiểu sự chọn lọc đặc trưng và tăng tính ngẫu nhiên để tăng khả năng tổng quát hóa của mô hình.
- Random Forest: Random Forest là một thuật toán học máy dựa trên việc xây dựng một tập hợp các cây quyết định ngẫu nhiên. Mỗi cây quyết định đóng góp vào việc đưa ra dự đoán và kết quả cuối cùng được tính toán bằng cách lấy giá trị trung bình hoặc phiếu bầu từ các cây con.
- Ensemble Learning: Ensemble Learning là một phương pháp kết hợp nhiều mô hình học máy để tạo ra một dự đoán chung. Các mô hình trong ensemble có thể là các thuật toán khác nhau hoặc các phiên bản khác nhau của cùng một thuật toán. Ensemble Learning có thể giúp cải thiện độ chính xác và khả năng tổng quát hóa của mô hình

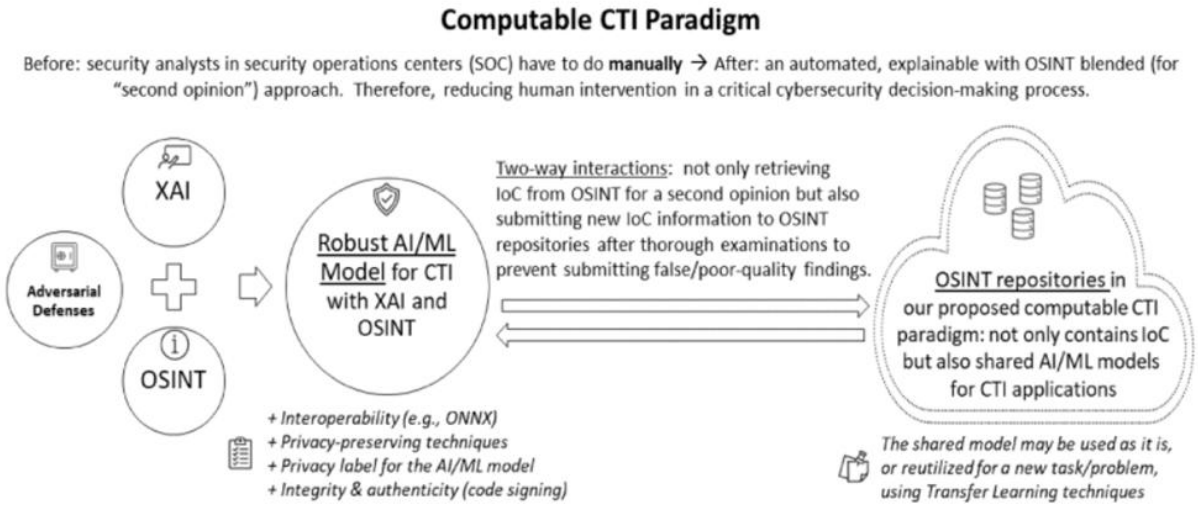
Chương 3: Phương pháp luận và thiết kế hệ thống

3.1 Mô hình chia sẻ CTI

Theo kiến thức của chúng tôi, chưa có nghiên cứu nào nhấn mạnh tầm quan trọng của chia sẻ mô hình AI/ML cho CTI. Hơn nữa, các nền tảng CTI hiện có MISP, NC4 CTX/Soltra Edge, ThreatConnect, AlienVault, IBM X-Force Exchange, Anomali, ThreatExchange, CrowdStrike, ThreatQuotient, EclecticIQ, CRITs, CIF v3, AlliaCERT) dường như không khuyến khích chia sẻ mô hình AI/ML cho CTI.

Do đó, nghiên cứu này hướng tới lấp đầy khoảng trống này bằng cách đề xuất mở rộng CTI có thể hoạt động, cụ thể là CTI có thể tính toán, một mô hình chia sẻ CTI mới. Chúng tôi xác định CTI có thể tính toán là cấp độ tiếp theo của CTI có thể hoạt động bằng cách mở rộng định nghĩa của Cơ quan An ninh mạng châu Âu (ENISA) sử dụng tiêu chí tính toán được của AI/ML . Hơn nữa, chúng tôi định nghĩa rằng mô hình chia sẻ CTI có thể tính toán khuyến khích chia sẻ mô hình AI hoặc ML của hệ thống CTI cho cộng đồng an ninh mạng.

Hình sau minh họa thiết kế khái niệm của CTI có thể tính toán. Các nguồn CTI khác nhau có sẵn trên thị trường và công khai cho cộng đồng như OSINT. Mô hình đề xuất của chúng tôi sử dụng hệ sinh thái OSINT để tăng cường kỹ thuật XAI bằng cách cung cấp ý kiến thứ hai từ IoC thu được từ OSINT, tránh định kiến tự động khi sử dụng AI/ML cho tự động hóa an ninh dựa trên CTI.



Hình 2. Mô hình CTI có thể tính toán

Gần đây, XAI trở thành đối tác quan trọng hỗ trợ người dùng và chuyên gia đưa ra quyết định quan trọng. XAI có thể hỗ trợ đưa ra quyết định nghiêm túc trong lĩnh vực y tế và an ninh khi xử lý các mối đe dọa mạng phức tạp/quan trọng. Bởi vì tính minh bạch đang được yêu cầu trong Quy định bảo vệ dữ liệu chung của Liên minh châu Âu (GDPR), nó trở nên quan trọng đối với các chuyên gia trong mọi ngành. Tuy nhiên, giải thích trong XAI cần phải thích ứng tùy theo bối cảnh và các yếu tố khác. Đạt được XAI đáng tin cậy vẫn là một trong những thách thức lớn mà các nhà nghiên cứu trong lĩnh vực này đang theo đuổi. Mục tiêu nghiên cứu của chúng tôi là đề xuất kết hợp XAI và OSINT để giải quyết vấn đề về sự tin cậy.

3.2. Các đặc trưng (Features)

Chúng tôi phân tích các tập dữ liệu bằng cách tính toán entropy sử dụng hàm Shannon như là đặc trưng đầu tiên của mô hình.Sự biến động về entropy có thể chỉ ra số lượng từ khóa truy vấn ngẫu nhiên duy nhất gia tăng trong truy vấn DNS, thường quan sát trong tình huống nguy hiểm như cuộc tấn công Kaminsky. Sau đó, chúng tôi mở rộng phép đo thống kê sử dụng entropy tương đối (RE) thông qua khoảng cách Kullback–Leibler.

$$H(X) = -\sum_{i \in X} P(i) \log_2 P(i) \tag{1}$$

$$DKL(P||Q) = \sum_i p_i \log \left(\frac{p_i}{q_i} \right) \tag{2}$$

Trong đó, Q là phân phối cơ sở tính toán trên dữ liệu hợp lệ (tên miền Alexa top 1 triệu hoặc tập dữ liệu 10 botnet), và P là phân phối mục tiêu (tức tên miền trong nhật ký truy vấn DNS cần kiểm tra).

Đặc trưng thứ hai của mô hình là RE-Alexa, đo khoảng cách (hay độ tương đồng) giữa tên miền cần kiểm tra với phân phối unigram của tên miền Alexa. Đặc trưng thứ ba là Min-RE-Botnets. Ở đây, chúng tôi tính toán giá trị RE của một tên miền nghi ngờ với từng tập dữ liệu botnet và lấy giá trị nhỏ nhất làm Min-RE-Botnets.

Được lấy cảm hứng từ công trình của Sharifnya , đặc trưng thứ tư trong mô hình là giá trị bán kính thông tin (IRad), tính toán bằng hàm phân kỳ Jensen-Shannon (3). Hàm này khái quát hóa phân kỳ Jensen-Shannon để so sánh nhiều phân phối xác suất hơn hai phân phối. Mô hình sử dụng hàm này để tính toán khoảng cách của tên miền mục tiêu với các tập dữ liệu botnet.

$$JSD\pi_1,\ldots,\pi_n(P_1,\ldots,P_n) = H\left(\sum_{i=1}^{n}\pi_i P_i\right) - \sum_{i=1}^{n}\pi_i H(P_i) \tag{3}$$

Đặc trưng tiếp theo là độ dài chuỗi ký tự của tên miền (CharLength). Đây là đặc trưng phù hợp cho việc phát hiện botnet DGA bởi vì một số thuật toán DGA trong tập dữ liệu của chúng tôi cho thấy độ dài ký tự tương tự, đặc trưng cho tính chất duy nhất của những tên miền được tạo ngẫu nhiên này.

Tiếp theo, một đặc trưng mới được sinh ra sử dụng thuật toán cây quyết định (TreeNewFeature). Ở đây, chúng tôi kết hợp các đặc trưng entropy, RE-Alexa, Min-RE-Botnets và CharLength sử dụng cây quyết định và sử dụng chúng để huấn luyện mô hình dự báo. Chúng tôi xây dựng cây quyết định sử dụng các đặc trưng này và sử dụng kết quả dự báo làm đặc trưng mới.

Đặc trưng cuối cùng là điểm uy tín Alexa (ReputationAlexa). Cách tiếp cận này lấy cảm hứng từ công trình của Zhao. Ở đây, chúng tôi sử dụng tên miền Alexa Top 1 triệu để tạo ma trận trọng số tính toán giá trị uy tín của tên miền.

$$W_{N-gram}(i) = \log_{10}\left(\sum_{i=1}^{n} C_{N-gram}(i)\right) \tag{4}$$

Trong đó, W là ma trận trọng số dùng để tính toán điểm uy tín và CN−gram là tần suất n-gram ký tự. Khi tính toán điểm uy tín của tên miền mục tiêu, trước tiên chúng tôi rút trích số lượng token từ tên miền đó sử dụng cấu trúc số liệu n-gram ký tự. Việc này tương đương với sinh ma trận tài liệu-thuật ngữ sử dụng Alexa Top 1 triệu.

3.3 Kết hợp XAI và OSINT

Nghiên cứu này áp dụng bốn kỹ thuật XAI hiện có (ANCHOR, Phép diễn giải cục bộ cho mô hình bất khả tri - LIME, Shapley Additive exPlanations - SHAP và giải thích phản biện và đề xuất cách tiếp cận của chúng tôi (kết hợp XAI và OSINT) để sản xuất giải thích ý kiến thứ hai.

Chúng tôi khai thác phương pháp SHAP để trình bày giải thích toàn cảnh dựa trên giải thích tính tương quan của đặc trưng lấy cảm hứng từ lý thuyết trò chơi. SHAP dựa trên giá trị Shapley lý thuyết trò chơi. Chúng tôi tập trung sử dụng cách tiếp cận mô hình bất khả tri để đạt tự do lớn hơn trong việc sử dụng thuật toán nâng cao cho mô hình phân loại. Do đó, chúng tôi áp dụng KernelExplainer để ước lượng giá trị Shapley dựa trên mô hình địa phương cho giải thích SHAP. Ngoài ra, chúng tôi sử dụng biểu đồ lực của SHAP để cung cấp giải thích cục bộ.

Phương pháp XAI tiếp theo triển khai trong nghiên cứu này là LIME. LIME huấn luyện các mô hình địa phương để giải thích dự báo/phân loại cụ thể. Nó cung cấp giải thích cục bộ, giải thích kết quả phân loại cụ thể của mô hình hộp đen. Do đó, người dùng sẽ hiểu tại sao hệ thống CTI lại phân loại tên miền bị nghi ngờ thành tên miền DGA hợp pháp hoặc botnet.

Tiếp theo, chúng tôi áp dụng ANCHORS, cải tiến của LIME, để dự đoán cách một mô hình sẽ hoạt động với ít nỗ lực hơn và độ chính xác cao hơn. ANCHORS là một phương pháp học dựa trên quy tắc, giải thích bằng cách đơn giản hóa. Chúng tôi trông chờ vào một lời giải thích được thể hiện dưới dạng các quy tắc IF-THEN dễ hiểu từ phương pháp này. Kiểu biểu thức này có thể thuận tiện hơn để giải thích hành vi của mô hình: tại sao hệ thống CTI lại quyết định một tên miền là miền DGA của botnet hoặc tại sao hệ thống CTI lại phân loại tên miền có vẻ đáng ngờ này là một tên miền hợp pháp? Chúng tôi đã sử dụng Alibi để triển khai ANCHORS trong hệ thống CTI của mình.

Tiếp theo, chúng tôi áp dụng cách giải thích phản thực tế, bổ sung khả năng giải thích bằng một ví dụ. Chúng tôi đã sử dụng Công cụ What-If để triển khai chức năng này, do đó cho phép trực quan hóa để làm nổi bật điểm dữ liệu phản thực gần nhất (nếu một tên miền hợp pháp được chọn thì tên miền DGA botnet gần nhất sẽ được hiển thị và ngược lại). Công cụ này sẽ cho phép các nhà phân tích an ninh mạng phát hiện những thay đổi tối thiểu về giá trị của các tính năng để khiến hệ thống CTI tạo ra các kết quả phân loại khác nhau. Do đó, hệ thống CTI có thể nhận được nhiều sự tin tưởng hơn từ người dùng vì họ hiểu được lời giải thích.

Đối với ý kiến thứ hai, chúng tôi đã sử dụng hai nguồn OSINT (Google Safe Browser và OTX AlienVault). Chúng tôi đã gửi truy vấn giao diện lập trình ứng dụng (API) tới các nguồn này để truy xuất nhận xét/báo cáo về miền bị nghi ngờ được đề cập. Chúng tôi đã hợp nhất thông tin này với đầu ra của mô hình DGA botnet của chúng tôi như một ý kiến thứ hai. IoC tổng hợp từ OSINT để xác nhận kết quả phân loại và đầu ra của mô hình AI/ML có thể được gửi đến kho lưu trữ OSINT sau

khi các chuyên gia kiểm tra kỹ lưỡng nhằm ngăn chặn việc gửi kết quả sai/kém chất lượng. Việc gửi thông tin IoC mới tới kho lưu trữ OSINT cần hết sức thận trọng, vì nó có thể cố ý hoặc vô ý là cách có thể gây ra sự cố với các trình phát hiện khác dựa vào OSINT đó và gây ảnh hưởng xấu đến bất kỳ quy trình đào tạo nào. Do đó, CTI có thể tính toán có thể nâng cao cộng đồng OSINT cho IoC về các mối đe dọa mới

CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM, PHÂN TÍCH – ĐÁNH GIÁ

Ở chương này chúng tôi tiến hành tạo môi trường, cài đặt và đưa ra các tiêu chí đánh giá về mức độ hiệu quả của mô hình.

4.1. Môi trường thực nghiệm

4.1.1 Tài nguyên

Máy ảo vmware:

- Hệ điều hành: kali-linux-2023.4-vmware-amd64
- Ổ cứng: HDD 60GB

Môi trường phát triển:

- Trình soạn thảo: Visual Studio Code.
- Ngôn ngữ lập trình: Python3.
- Thư viện sử dụng: numpy, pandas, mathplotlib, sklearn, ...

4.1.2 Tập dữ liệu

Việc sử dụng thuật toán ML để phát hiện lưu lượng DNS độc hại yêu cầu dữ liệu thực tế chính xác cho cả việc đào tạo mô hình và đánh giá độ chính xác. Đối với thử nghiệm đầu tiên trong nghiên cứu của chúng tôi, chúng tôi đã sử dụng Alexa Top 1M (1.000.000 tên miền) và 803.333 tên miền của mười họ DGA botnet được sử dụng trong : Conficker, Cryptolocker, Goz, Matsnu, New_Goz, Pushdo, Ramdo, Rovnix, Tinba , Zeus. Sau đó, chúng tôi sử dụng 998.503 tên miền của 55 họ DGA từ Netlab 360.

Bảng 2. Tập dữ liệu với tổng 55 họ nhà DGA

DGA Family	Sample content of the dataset
abcbot	fuorhtpsx.tk
antavmu	5f474370.com
bamital	a9d68c6203f04de3265bb8c7584e476b.info
banjori	igxbtallulahavaw.com
bigviktor	callleastrace.fans
blackhole	owrfrxrmiewneegp.ru
ccleaner	ab2ec634a79.com
chinad	797nkllqgz9x7x28.com
conficker	dxtpafsyjcw.cc
copperstealer	dfd886470ec28240.com
cryptolocker	xeuskjcythfllwh.org
dircrypt	rvonetvypqacbsa.com
dyre	y243c1001a327885f71ee399229ef82609.cc
emotet	qijfcnekvhwvcgkg.eu
enviserv	33aaf2199f.net
feodo	mgcdlsidwsdnolwzyz.ru
flubot	oupxbsfpvukowup.cn
fobber	ylbphjjdjs.com
gameover	115vvgdobj3uf1jq8gi174q2t7.net
gspy	cc9cf6ae3922d07d.info
kfos	help-google.tw
locky	gcqbsfpxkhqf.tf
madmax	www.k3bdsbsa3k.net
matsnu	dishcow-catcondition.com
mirai	cmhewcvopvno.online
monerominer	5c95f79304b49.org
murofet	niqlkgqytoirnou.org
mydoom	hrsrrarpsr.net
necro	vlxdqiwafmbashxa.viewdns.net
necurs	hjvtlavpidi.su
ngioweb	subozirion-multirusenelike.com
nymaim	yowgbazj.pw
omexo	8f86373028729e6497f00487d7775f81.net
padcrypt	efddmacncmndodcn.website
proslikefan	pbeuiykocu.ru
pykspa	zzzkdgn.org
qadars	7g9ijc9e3why.net
qakbot	pnhwjybkdixb.com
ramnit	nnnfyqtv.com
ranbyus	gwoukqdssttrhg.me
rovnix	o4cwfxophfp7iiq4zs.biz
shifu	gemmpbt.eu
shiotob	sifulj2yl1g.com
simda	pufybyg.info
suppobox	coriandertimothyson.net
symmi	weaxabudodine.ddns.net
tempedreve	ozmlglgb.info
tinba	wddyvorijopl.ru
tinynuke	dedc87d2095576f2842c7be426613667.com
tofsee	eaieaih.biz
tordwm	dab53527.top
vawtrak	mxubexcvqvc.com
vidro	wwcdjkdsg.dyndns.org
virut	jawukx.com
xshellghost	texcrgluvmrgr.com

4.1.3 Thí nghiệm

Chúng tôi đã tiến hành ba thí nghiệm như sau:

- Đầu tiên, sử dụng nhiều thuật toán ML được giám sát để so sánh kết quả nhằm chọn ra thuật toán có độ chính xác tốt nhất. Ở đây, năm thuật toán (naive Bayes, logistic regression, extra tree, random forest, and ensemble learning) đã được tính toán bằng Scikit-Learn.
- Tiếp theo là so sánh mô hình rừng ngẫu nhiên với mô hình mới nhất trước đó.
- Để kiểm tra hiệu suất của mô hình phát hiện DGA botnet trong việc xử lý các cuộc tấn công đối nghịch khắc nghiệt, chúng tôi đã tiến hành đánh giá độ mạnh của bộ phân loại của chúng tôi trước ba cuộc tấn công DGA tiên tiến (CharBot, DeepDGA, MaskDGA) và so sánh với bốn mô hình deep learning (Endgame, CMU, NYU, MIT).

Số liệu đánh giá được đưa ra trong (5), trong đó TP, TN, FP, FN lần lượt là dương tính thực, âm tính thực, dương tính giả và âm tính giả. Bảng 3 tóm tắt các biến/tham số được sử dụng trong mô hình rừng ngẫu nhiên.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

Bảng 3: Tóm tắt các tham số được sử dụng trong mô hình Random Forest Model

Parameters	Values
bootstrap	True
ccp_alpha	0.0
class_weight	None
criterion	gini
max_depth	None
max_features	auto
max_leaf_nodes	None
max_samples	None
min_impurity_decrease	0.0
min_impurity_split	None
min_samples_leaf	1
min_samples_split	2
min_weight_fraction_leaf	0.0
n_estimators	1500
n_jobs	40
oob_score	False
random_state	1
verbose	1
warm_start	False

4.2 Kết quả thí nghiệm

4.2.1 So sánh độ chính xác của thuật toán ML

Kết quả thí nghiệm của chúng tôi được thể hiện ở Bảng 4 và 5. Nhìn chung, mô hình rừng ngẫu nhiên đạt độ chính xác cao nhất, tiếp theo là thuật toán cây phụ. Trong đó naive bayes luôn cho thấy hiệu suất thấp nhất trong số các thuật toán được so sánh. Độ chính xác cao nhất (96,2%) đạt được bằng cách sử dụng rừng ngẫu nhiên với tất cả bảy tính năng. Ba tính năng thiết yếu hàng đầu là CharLength, ReputationAlexa và TreeNewFeature.

BẢNG 4 Độ chính xác của Mô hình ML khi sử dụng kết hợp nhiều tính năng khác nhau (1/2)

Features	3 features	4 features	4 features	4 features	4 features	5 features	5 features	5 features
- CharLength	√	√	√	√	√	√	√	√
- TreeNewFeature	√	√	√	√	√	√	√	√
- ReputationAlexa	√	√	√	√	√	√	√	√
- RE-Alexa	-	√	-	-	-	√	√	√
- Min-RE-Botnets	-	-	√	-	-	√	-	-
- Entropy	-	-	-	√	-	-	√	-
- IRad	-	-	-	-	√	-	-	√
Logistic Regression	89.5%	89.8%	89.5%	90.6%	90.0%	90.1%	90.7%	90.0%
Random Forest	92.7%	94.6%	95.1%	94.4%	94.8%	95.7%	95.3%	95.2%
Naive Bayes	83.2%	83.5%	82.7%	82.5%	82.5%	82.9%	82.9%	82.9%
Extra Tree	92.7%	94.3%	94.8%	94.2%	94.5%	95.6%	95.2%	95.1%
Ensemble	91.7%	93.6%	94.4%	94.6%	94.1%	94.7%	94.5%	94.1%

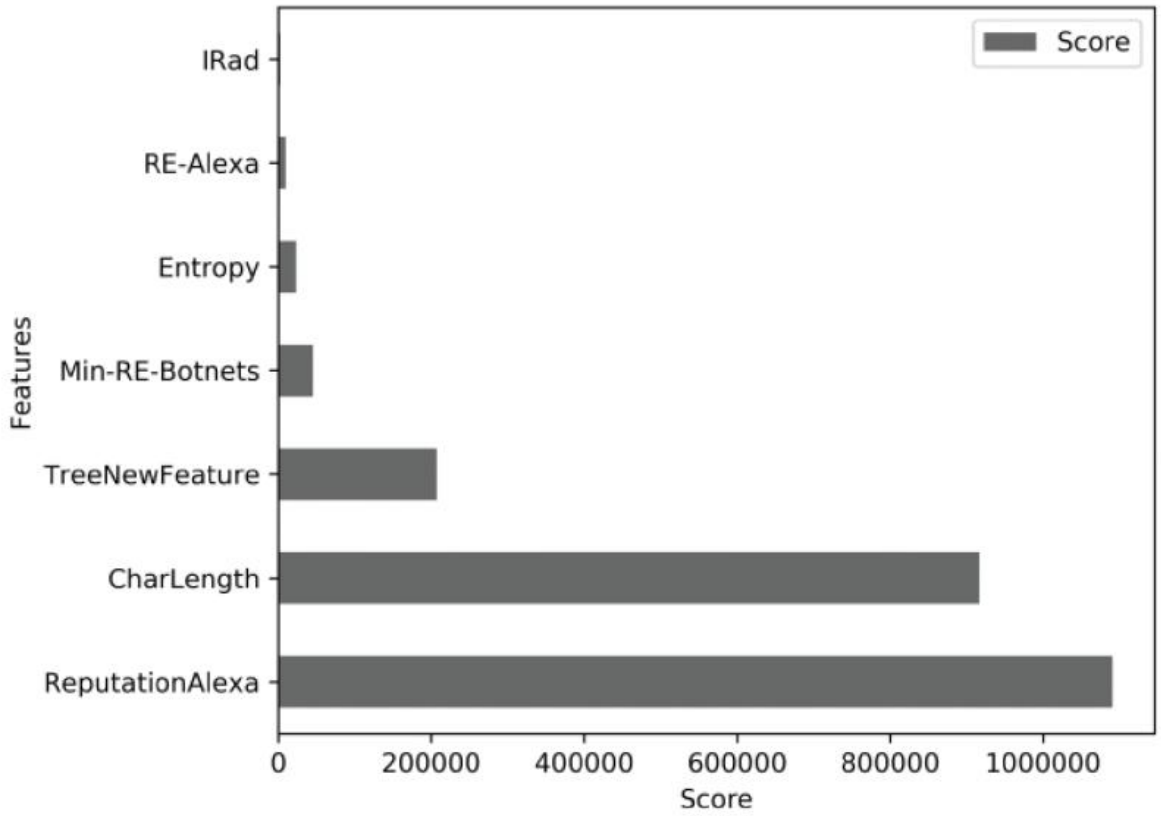
BẢNG 5 Độ chính xác của Mô hình ML khi sử dụng kết hợp nhiều tính năng khác nhau (2/2)

Features	5 features	5 features	5 features	6 features	6 features	6 features	6 features	7 features
- CharLength	√	√	√	√	√	√	√	√
- TreeNewFeature	√	√	√	√	√	√	√	√
- ReputationAlexa	√	√	√	√	√	√	√	√
- RE-Alexa	-	-	-	√	√	√	-	√
- Min-RE-Botnets	√	√	-	√	√	-	√	√
- Entropy	√	-	√	√	-	√	√	√
- IRad	-	√	√	-	√	√	√	√
Logistic Regression	91.3%	90.9%	90.8%	91.3%	90.5%	90.7%	91.4%	91.3%
Random Forest	95.9%	95.5%	95.6%	96.1%	95.8%	95.8%	96.1%	96.2%
Naive Bayes	81.9%	81.9%	81.9%	82.2%	82.2%	82.3%	81.5%	81.5%
Extra Tree	95.8%	95.4%	95.5%	96.1%	95.7%	95.8%	96.1%	96.2%
Ensemble	95.2%	94.6%	94.8%	95.2%	94.6%	94.6%	95.1%	95.0%

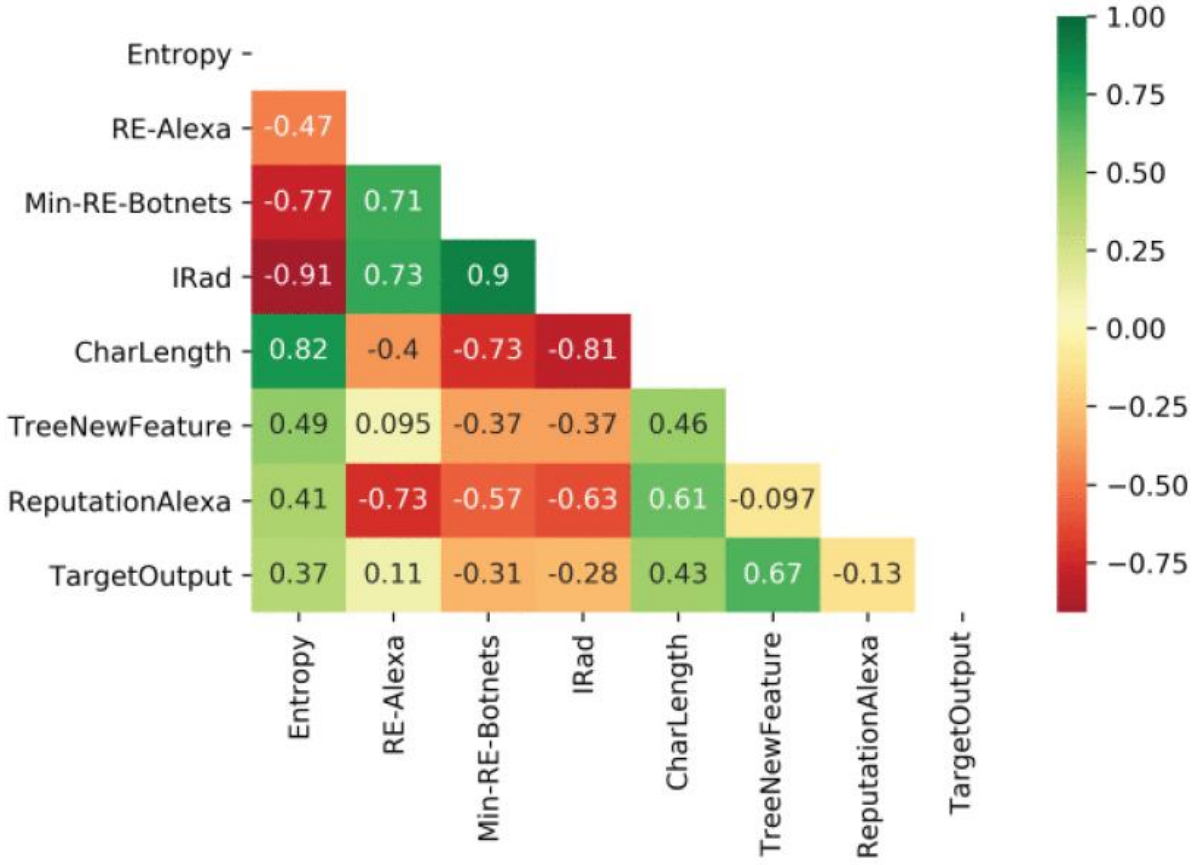
Chúng tôi đã phân tích tất cả các đặc điểm bằng cách sử dụng các bài kiểm tra thống kê để chọn ra các đặc điểm có mối quan hệ chặt chẽ nhất với biến đầu ra. Hình 3 cho thấy kết quả của phép thử Chi bình phương lựa chọn đơn biến. Các tính năng ReputationAlexa, CharLength và TreeNewFeature có mối quan hệ cao nhất với đầu ra của lớp. Sau đó, chúng tôi đã nghiên cứu xem các tính năng có liên quan với nhau như thế nào bằng cách sử dụng ma trận tương quan. Như được hiển thị trong Hình 4, các tính năng TreeNewFeature, Char-Length, Entropy và RE-Alexa có mối tương quan tích cực với đầu ra và có thể quan sát thấy mối tương quan nghịch đối với các tính năng Min-RE-Botnets và IRad. Hơn nữa, chúng tôi đã thực hiện phân tích tầm quan trọng của tính năng để chấm điểm từng tính năng trong mô hình đề xuất của chúng tôi.

Như được hiển thị trong Hình 5, các tính năng TreeNewFeature, ReputationAlexa và CharLength đạt được điểm cao nhất, điều này cho thấy rằng các tính năng này rất cần thiết cho biến đầu ra.

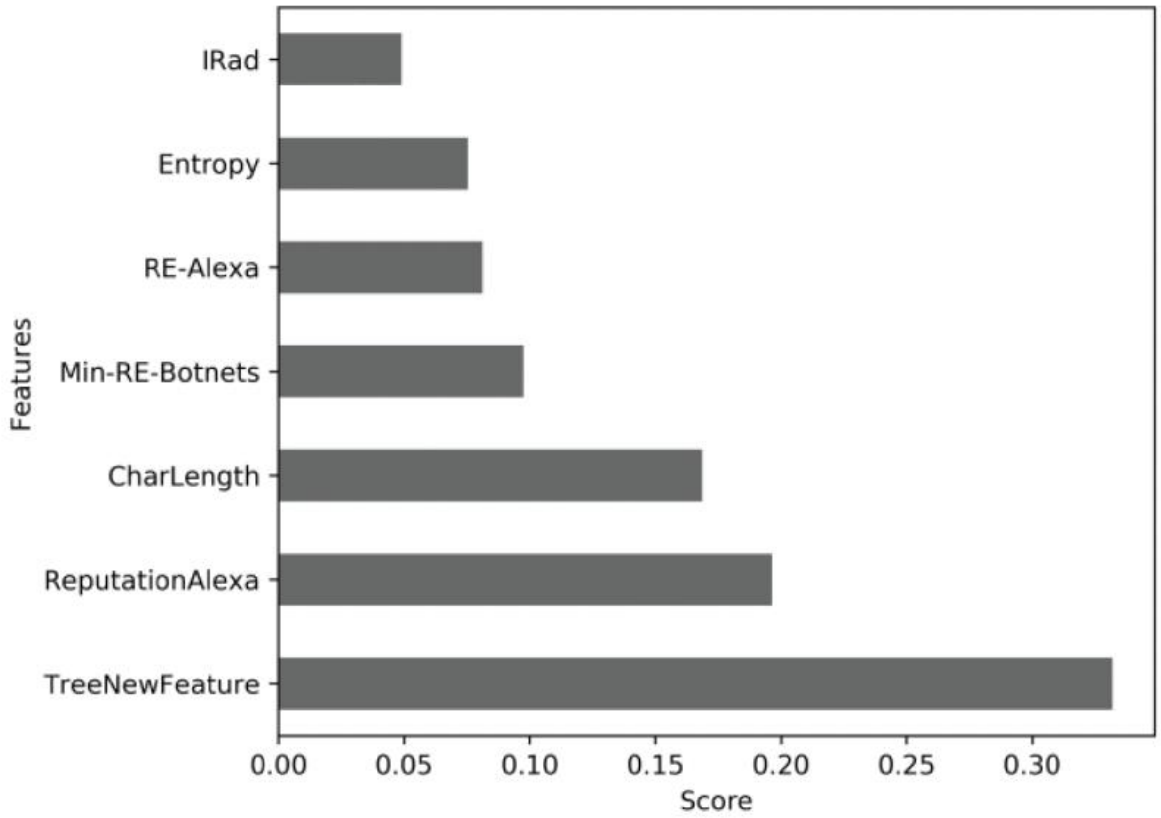
Hình 3. Kết quả phép thử chi bình phương



Hình 4. Kết quả phân tích ma trận tương quan.



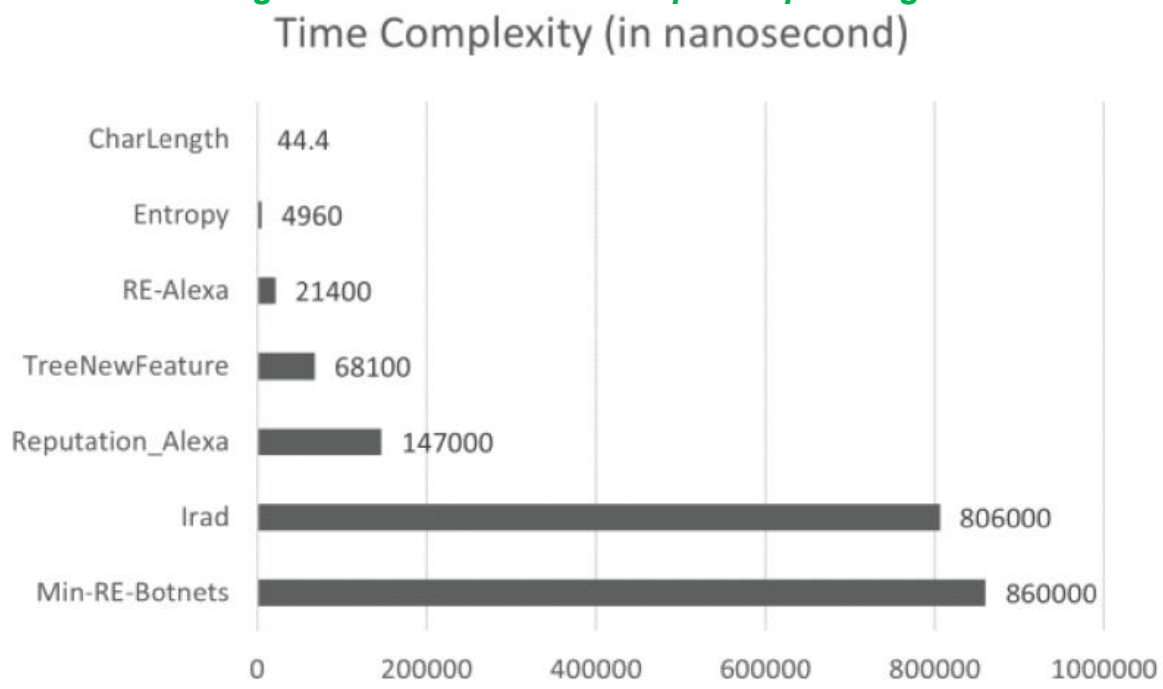
Hình 5. Kết quả phân tích các đặc trưng quan trọng



4.2.2 Độ phức tạp về thời gian để tính toán các tính năng

Bộ phân loại ML sử dụng cách tiếp cận dựa trên thống kê yêu cầu tính toán để tính toán các tính năng của chúng. Hình 6 cho thấy chi phí tính toán các phương pháp của chúng tôi xét về độ phức tạp về thời gian để tính toán các tính năng cần thiết trong mô hình. Min-RE-Botnets và IRad yêu cầu thời gian tính toán dài hơn các tính năng khác, vì phương trình (2) và (3) mang lại hệ quả là độ phức tạp thời gian tuyến tính O(n), với n= số lượng họ DGA (55 họ trong thí nghiệm). Điều này sẽ trở thành bất lợi khi số lượng họ DGA tăng lên. Tuy nhiên, tính năng ReputationAlexa không yêu cầu tính toán nhiều vì việc chuẩn bị chỉ cần được thực hiện một lần trong bước đào tạo mô hình: đọc tất cả các miền từ Alexa và sau đó học từ điển từ vựng của n-gram 3 đến 5 ký tự để tạo trọng số ma trận.

HÌNH 6. Thời gian tính toán để chuẩn bị các đặc trưng cho mô hình.



4.2.3 So sánh với nghiên cứu trước đó

Bảng 6 cung cấp sự so sánh giữa mô hình rừng ngẫu nhiên được đề xuất của chúng tôi và công việc trước đó (Hoàng và Vũ), sử dụng cùng một cách tiếp cận (dựa trên các đặc điểm thống kê) và cùng một thuật toán rừng ngẫu nhiên. Sử dụng cùng cài đặt bộ dữ liệu, cho tất cả các thử nghiệm, mô hình của chúng tôi mang lại tỷ lệ phát hiện tốt hơn (với độ chính xác trung bình là 98,9%), mặc dù phương pháp của chúng tôi chỉ sử dụng bảy đặc trưng so với phương pháp của họ, phụ thuộc vào 24 đặc trưng.

BẢNG 6 So sánh độ chính xác/Tỷ lệ phát hiện

Datasets	Our Random Forest Model (with only 7 features)	Hoang and Vu [7] (with 24 features)
39 DGA families	99.3%	83.8%
25 DGA families	99.7%	98.0%
10 DGA families	97.2%	83.1%
4 DGA families	99.3%	75.0%
average =	*98.9%*	*85.0%*

Những kết quả này đưa ra bằng chứng rõ ràng về lợi thế của việc sử dụng tiềm năng bảy đặc trưng được đề xuất trong bài viết của chúng tôi để mang lại hiệu suất phát hiện DGA botnet thỏa đáng.

4.2.4 Đánh giá độ bền

Đầu tiên, chúng tôi đã kiểm tra hiệu suất của mô hình rừng ngẫu nhiên với bảy đặc trưng bằng cách sử dụng bộ dữ liệu thực tế bao gồm tên miền của Alexa và 55 họ DGA (tổng cộng 1.998.502 tên miền). Như được hiển thị trong Bảng 7, các mô hình học sâu dựa trên ký tự tạo ra độ chính xác cao hơn một chút (~99,0%) so với mô hình của chúng tôi (độ chính xác 96,3%).

BẢNG 7 Khả năng chống lại các cuộc tấn công DGA tiên tiến nhất

Datasets	OUR MODEL	Character-Based Deep Learning Model [8]			
		Endgame [35]	CMU [36]	NYU [37]	MIT [38]
- Alexa Top 1M and 55 DGA families	96.3%	98.9%	99.0%	98.9%	99.0%
- CharBot, MaskDGA, and DeepDGA attacks	44.2%	35.8%	30.5%	38.4%	29.7%
- CharBot attack	17.4%	15.0%	10.9%	9.1%	10.0%
- MaskDGA attack	19.7%	27.4%	17.1%	16.3%	30.9%
- DeepDGA attack	45.8%	36.7%	31.6%	40.0%	30.3%

Tuy nhiên, đánh giá độ mạnh với các cuộc tấn công CharBot, MaskDGA và DeepDGA (tổng cộng 394.000 tên miền) đưa ra bằng chứng cho thấy mô hình của chúng tôi cung cấp khả năng phòng thủ tốt hơn trước cả ba cuộc tấn công DGA (độ chính xác 44,2%). Đánh giá chống lại các cuộc tấn công DGA riêng lẻ cho thấy mô hình của chúng tôi có độ bền tốt hơn trước các cuộc tấn công CharBot và DeepDGA, ngoại trừ cuộc tấn công MaskDGA.

Những kết quả này cho thấy những ưu điểm của mô hình của chúng tôi được sử dụng để phát hiện DGA botnet trong việc xử lý các cuộc tấn công DGA khắc nghiệt, trong đó một cuộc tấn công DGA mới có thể làm giảm đáng kể độ chính xác của bộ phân loại DGA lên tới độ chính xác chỉ 9,1% (trong trường hợp của NYU). mô hình được thử nghiệm với cuộc tấn công CharBot). Xu hướng này cũng tương tự như các nghiên cứu trước đây.

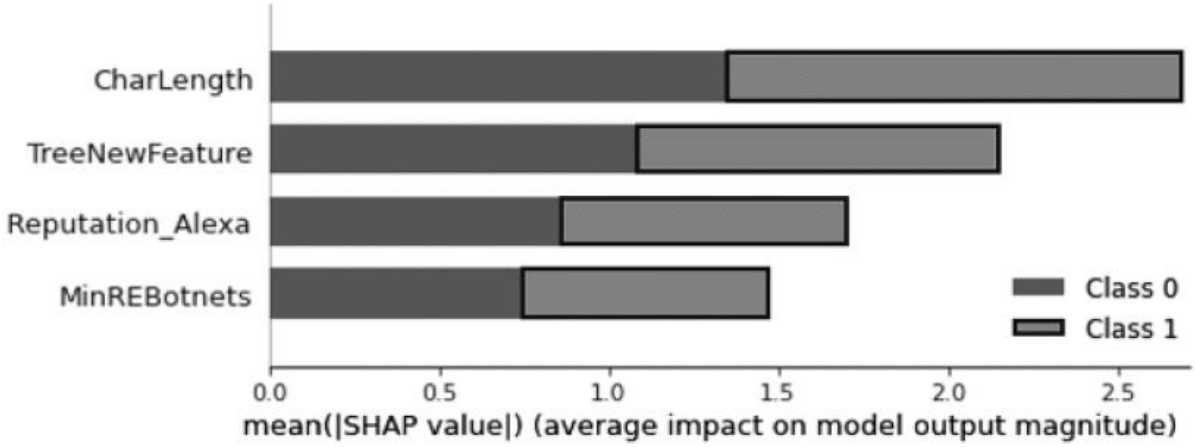
4.2.5 Cơ chế chia sẻ trong CTI có thể tính toán

Chúng tôi đã xác định một số giao thức có tiềm năng triển khai các mô hình AI/ML chia sẻ, chẳng hạn như bộ chứa docker, gốc (phụ thuộc vào các công cụ được sử dụng, ví dụ: joblib cho Python); Định dạng trao đổi mô hình dự đoán dựa trên PMML/XML; và trao đổi mạng thần kinh mở (ONNX), tiêu chuẩn mở cho khả năng tương tác ML. Chúng tôi đã tuần tự hóa mô hình ML cuối cùng của mình để phát hiện DGA botnet bằng cách sử dụng các phương pháp tiếp cận joblib của ONNX và Python. Việc tuần tự hóa và giải tuần tự hóa có thể diễn ra suôn sẻ. Mặc dù kích thước tệp của mô hình được đào tạo có thể trở nên lớn khi dữ liệu đào tạo khổng lồ, việc chia sẻ mô hình được đào tạo/sẵn sàng sử dụng sẽ rất thuận tiện cho những người khác cần phân tích lưu lượng DGA của botnet mà không phải chịu gánh nặng xây dựng và đào tạo mô hình.

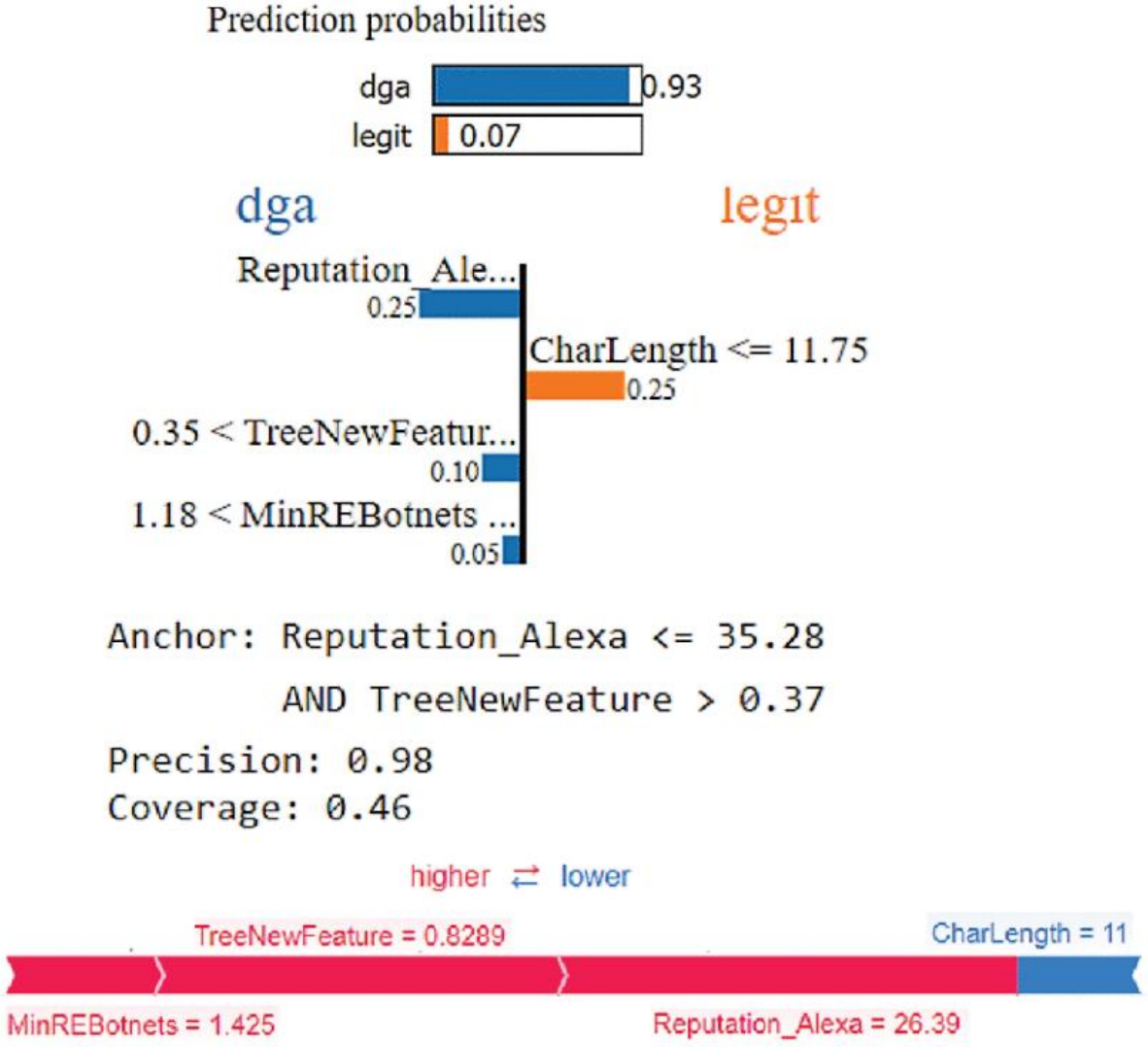
4.2.6 Kết hợp XAI và OSINT

Đầu tiên, hệ thống CTI của chúng tôi hiển thị phần giải thích tổng thể về mô hình (Hình 7). Mô hình của chúng tôi coi độ dài ký tự (character length) là đặc trưng chính trong việc nhận dạng tên miền DGA của botnet. Do đó, miền có số lượng ký tự quá dài có xu hướng là tên miền DGA botnet, điều này đúng, dựa trên tập dữ liệu thực tế của chúng tôi. Để tin tưởng vào mô hình, người dùng phải hiểu mô hình đó tốt ở điểm nào và khi nào mô hình có thể sai. Chúng tôi cung cấp hình ảnh trực quan (Hình 8), cho phép nhà phân tích an ninh mạng biết cách phân loại sai ở đâu, chẳng hạn như khi có một tên miền hợp pháp tồn tại nhưng mô hình lại phân loại nó là tên miền DGA của botnet. Mặc dù mô hình có độ chính xác cao nhưng nếu nó phân loại một tên miền hợp pháp nổi tiếng (chẳng hạn như google.com) là botnet DGA thì điều đó là không thể chấp nhận được.

HÌNH 7. Sơ đồ tóm tắt giải thích toàn cục SHAP.



HÌNH 8. Giải thích cục bộ: LIME hiển thị biểu đồ dự đoán, ANCHORS hiển thị Quy tắc IF-THEN và SHAP cung cấp biểu đồ lực.



Chúng tôi cung cấp các giải thích cục bộ hoặc giải thích một đầu ra quyết định duy nhất để trình bày ý tưởng đơn giản hóa logic về lý do tại sao mô hình đưa ra quyết

định đó. Chúng tôi đưa ra một ví dụ về lý do tại sao hệ thống CTI xác định rằng một tên miền trông bình thường (trong đó số lượng ký tự không quá nhiều) được phân loại là tên miền DGA của botnet (Hình 8). LIME hiển thị đồ thị dễ hiểu. Mặc dù độ dài ký tự ngắn khiến nó trông giống như một tên miền hợp pháp, nhưng cách tính điểm danh tiếng so với các tên miền Top 1M của Alexa lại đưa ra quyết định ngược lại. ANCHORS đưa ra lời giải thích tương tự, nhưng ở định dạng quy tắc IF-THEN. Hơn nữa, SHAP hiển thị cách mỗi giá trị của tính năng buộc phải đưa ra quyết định về kết quả phân loại DGA hợp pháp hoặc botnet. Lời giải thích ngược lại rất hữu ích để khiến các nhà phân tích an ninh mạng tin tưởng vào mô hình của chúng tôi bằng cách cung cấp các ví dụ: hai tên miền có đặc điểm tương tự nhưng được phân loại thành các lớp khác nhau; một cái là botnet DGA và cái còn lại là tên miền hợp pháp.

Sau khi hiển thị các giải thích SHAP, LIME, ANCHORS và phản thực tế, chúng tôi tiếp tục đưa ra ý kiến thứ hai (Hình 9) bằng cách tích hợp các kết quả truy vấn API từ hai nguồn OSINT (Google Safe Browser và OTX AlienVault). Do đó, chúng tôi xác nhận rằng lời giải thích hợp lý và ý kiến thứ hai (bằng cách triển khai kết hợp XAI và OSINT) là chìa khóa để thiết lập độ tin cậy khi sử dụng mô hình AI/ML chung.

HÌNH 9. Ý kiến thứ hai từ việc kết hợp XAI và OSINT: tích hợp kết quả truy vấn API từ Google Safe Browser và OTX AlienVault.

Trong nghiên cứu của chúng tôi về phát hiện DGA botnet, sai lệch tự động hóa đề cập đến một hành động mà nhà phân tích an ninh mạng không bao giờ nghi ngờ về kết quả quyết định của mô hình AI/ML, bất kể chúng là gì. Chẳng hạn như khi mô hình phát hiện sai tên miền là DGA botnet độc hại và nhà phân tích an ninh mạng quá tin tưởng nó. Chúng tôi nhấn mạnh rằng việc kết hợp XAI và OSINT có thể giải quyết xu hướng tự động hóa thông qua ý kiến thứ hai.

Cờ giả trên mạng là chiến thuật của tin tặc nhằm đánh lừa hoặc đánh lừa các nỗ lực phân bổ và các cuộc tấn công mạng bí mật. Bằng cách kết hợp XAI và OSINT vào các

hệ thống CTI dựa trên AI/ML, các nhà phân tích an ninh mạng có một công cụ tiện dụng để so sánh mọi thông tin từ các nguồn OSINT, với kết quả của mô hình được lấy từ kho chia sẻ CTI (đo hai lần, cắt một lần bằng mô hình AI/ML để xác nhận thông tin OSINT). Mô tả này nêu bật lên tính hữu ích của sự kết hợp XAI và OSINT được đề xuất của chúng tôi đối với hiện tượng cờ giả trên mạng.

4.2.7 Ý nghĩa thực tế của CTI có thể tính toán

Giảm sự can thiệp của con người vào việc ra quyết định về an ninh mạng bằng cách sử dụng tính năng tự động hóa AI/ML sẽ giúp các nhà phân tích bảo mật trong môi trường trung tâm điều hành bảo mật giành chiến thắng trong cuộc chạy đua vũ trang chống lại các mối đe dọa mạng mới. Mô hình CTI có thể tính toán nhấn mạnh mô hình AI/ML mạnh mẽ với các kỹ thuật phòng thủ đối nghịch, đồng thời kết hợp XAI và OSINT để giải quyết xu hướng tự động hóa.

Mô hình CTI có thể tính toán cũng khuyến khích các cộng đồng an ninh mạng đóng góp kết quả phát hiện CTI được quản lý cẩn thận của họ để làm phong phú thêm dữ liệu IoC trong kho lưu trữ OSINT. API OSINT tích hợp các mô hình AI/ML để cho phép gửi thông tin về mối đe dọa mới tới cơ sở dữ liệu OSINT. Trong nghiên cứu điển hình về phát hiện DGA botnet, chúng tôi đã sử dụng API DirectConnect của OTX AlienVault để chứng minh việc gửi các phát hiện mới được xác nhận và xác thực, khi không có OSINT nào tồn tại cho tên miền DGA của botnet. Do đó, CTI có thể tính toán ngụ ý sự tương tác hai chiều: thu được lợi ích từ việc tổng hợp dữ liệu về mối đe dọa OSINT và đóng góp vào IoC của các mối đe dọa mới nhất để giải quyết các vectơ tấn công toàn cầu mới.

Đầu tiên, liên quan đến khả năng tương tác của mô hình AI/ML, chúng tôi đã trình bày cách có thể quản lý khả năng tương tác khi chia sẻ mô hình AI/ML bằng ONNX. Việc áp dụng tiêu chuẩn này sẽ loại bỏ rào cản bị khóa trên một nền tảng AI/ML. Việc chia sẻ các mô hình CTI theo tiêu chuẩn ONNX sẽ tiếp cận được nhiều đối tượng cộng đồng an ninh mạng hơn.

Thứ hai, quyền riêng tư của người dùng phải được bảo vệ vì việc chia sẻ mô hình diễn ra giữa những người dùng. Chúng tôi đề xuất áp dụng nhãn quyền riêng tư (mã màu: trắng, xanh lá cây, hổ phách và đỏ) liên quan đến các biện pháp liên quan đến quyền riêng tư và việc tuân thủ các quy định về quyền riêng tư trên các mô hình dùng chung. Các kỹ thuật bảo vệ quyền riêng tư khác nhau có thể được sử dụng khi các mô hình bao gồm lưu trữ, xử lý và truyền thông tin cá nhân.

Thứ ba, mô hình CTI có thể tính toán khuyến khích áp dụng thực tiễn ký mã để đảm bảo tính toàn vẹn và tính xác thực của mô hình AI/ML được chia sẻ. Sigstore, một dự án được công bố gần đây của Quỹ Linux nhằm thúc đẩy việc áp dụng ký mã hóa, có thể trở thành chất xúc tác cho việc áp dụng rộng rãi CTI có thể tính toán trong cộng đồng an ninh mạng và nguồn mở.

CHƯƠNG 5: DEMO CỦA NHÓM

5.1 Đọc dữ liệu

Đọc dữ liệu từ file dataset data_exported_7features.csv, gồm 1.803.332 tên miền với các đặc trưng.

```
#1 Read The Data: dataset/data_exported_7features.csv
Columns (19) have mixed types. Specify dtype option on import or set low_memory=False.
   Num  No    Domainname  Label   Entropy   REAlexa  REConficker  RECryptolocker     REGoz  ...    REZeus  MinREBotnets
0    0   0    google.com  legit  1.918296  2.255828     2.669310        2.527949  2.675985  ...  3.246443      2.154914
1    1   1  facebook.com  legit  2.750000  1.634639     1.847340        1.700624  1.866545  ...  2.398782      1.379820
2    2   2   youtube.com  legit  2.521641  2.020058     2.074149        2.004490  2.059402  ...  2.633640      2.004490
3    3   3     baidu.com  legit  2.321928  2.113081     2.266616        2.246880  2.264237  ...  2.826578      1.843343
4    4   4     yahoo.com  legit  1.921928  2.377796     2.658381        2.506954  2.661281  ...  3.197753      2.328499
```

```
InformationRadius  ClassificationResult   Result  CharLength  LabelBinary  TreeNewFeature  nGramReputation_Alexa
         0.824130                 legit  Correct          10            0        0.148104              77.921931
         0.687778                 legit  Correct          12            0        0.828927              94.942922
         0.738282                 legit  Correct          11            0        0.353117              88.405451
         0.785090                 legit  Correct           9            0        0.148104              47.560154
         0.838762                 legit  Correct           9            0        0.148104              55.724565
```

5.2 Tính toán và chuẩn bị dữ liệu với các đặc trưng được chọn

```
#2 Calculate / Prepare The Data (Features Selection)
0                              google.com
1                            facebook.com
2                             youtube.com
3                               baidu.com
4                               yahoo.com
                          ...
1803328    r3o3mt1q7qhld1mp4g2akzqs37.biz
1803329     b05q9rw9lv1d1aq5po08iyjn5.org
1803330     sgem711uuk2vmyl1qlrdymhvl.org
1803331     ozhujl16ayo6crwwdf7fxskdk.org
1803332     9hw0nq1p9binuc6jrifi1noiu.biz
Name: Domainname, Length: 1803333, dtype: object
[[10.          0.14810403 77.92193141  2.15491358]
 [12.          0.82892709 94.94292228  1.37982007]
 [11.          0.3531169  88.40545079  2.00448959]
 ...
 [29.          0.99047121 36.56844303  1.12638625]
 [29.          0.99047121 32.73987616  0.95187191]
 [29.          0.99047121 36.28514263  1.13532149]]
['legit' 'legit' 'legit' ... 'dga' 'dga' 'dga']
[0 0 0 ... 1 1 1]
```

5.3 Chuẩn bị các dữ liệu kiểm tra đào tạo

```
#3 Prepare the trainin testing data
[[1.10000000e+01 3.53116900e-01 3.41664814e+01 1.81234732e+00]
 [1.10000000e+01 3.53116900e-01 6.89474864e+01 1.81860871e+00]
 [3.10000000e+01 9.90471212e-01 4.97036874e+01 1.04156069e+00]
 [1.20000000e+01 2.27128863e-01 8.22146963e+01 1.54062419e+00]
 [1.60000000e+01 1.07095312e-01 5.02464045e+01 1.28498853e+00]
 [1.10000000e+01 8.28927094e-01 2.26827100e+01 1.35458825e+00]
 [3.30000000e+01 8.46070937e-01 2.02935760e+02 3.88658610e-01]
 [3.20000000e+01 6.67704770e-01 1.74053896e+02 5.88581876e-01]
 [1.30000000e+01 1.07095312e-01 8.62949710e+01 1.17374594e+00]
 [3.30000000e+01 6.67704770e-01 2.16338054e+02 6.70668906e-01]
 [1.40000000e+01 2.00584157e-03 6.88790676e+01 1.98824646e+00]
 [1.00000000e+01 2.27128863e-01 4.08003535e+01 1.41648223e+00]
 [2.30000000e+01 1.07095312e-01 1.47123101e+02 9.66270542e-01]
 [1.80000000e+01 1.07095312e-01 1.15950727e+02 1.06027767e+00]
 [1.60000000e+01 8.28927094e-01 3.77148954e+01 1.75855044e+00]
 [1.90000000e+01 1.07095312e-01 1.45554078e+02 1.11550547e+00]
 [1.90000000e+01 8.28927094e-01 4.74175795e+01 1.47766967e+00]
 [3.00000000e+01 9.90471212e-01 2.78771213e+01 9.20528443e-01]
 [1.10000000e+01 3.53116900e-01 5.19901971e+01 1.92682816e+00]
 [2.60000000e+01 9.67922957e-01 1.61899632e+02 7.16645334e-01]
 [1.10000000e+01 3.53116900e-01 1.58813819e+01 1.86495223e+00]
 [1.90000000e+01 1.07095312e-01 1.21131142e+02 1.26223179e+00]
 [2.20000000e+01 3.67645424e-01 1.95110763e+02 8.22933902e-01]
 [2.90000000e+01 9.90471212e-01 3.88150906e+01 9.34726131e-01]
 [1.70000000e+01 1.07095312e-01 1.07825879e+02 1.15203733e+00]
 [1.20000000e+01 2.27128863e-01 6.93105806e+01 1.61619693e+00]
 [1.30000000e+01 1.07095312e-01 6.50886861e+01 1.38485001e+00]
 [1.60000000e+01 2.27128863e-01 6.29913821e+01 1.20021057e+00]
 [2.50000000e+01 1.07095312e-01 1.62580141e+02 1.01202548e+00]
 [2.10000000e+01 2.27128863e-01 1.13812358e+02 1.32986507e+00]
 [2.40000000e+01 3.67645424e-01 1.36532064e+02 7.52688916e-01]
```

```
['dga' 'legit' 'dga' 'legit' 'dga' 'dga' 'dga' 'dga' 'legit' 'dga' 'legit'
 'legit' 'legit' 'legit' 'dga' 'legit' 'dga' 'dga' 'legit' 'dga' 'dga'
 'legit' 'legit' 'dga' 'legit' 'legit' 'legit' 'dga' 'legit' 'legit' 'dga'
 'legit' 'legit' 'legit' 'legit' 'legit' 'legit' 'legit' 'legit' 'legit'
 'legit' 'legit' 'legit' 'dga' 'legit' 'legit' 'dga' 'legit' 'legit' 'dga'
 'dga' 'dga' 'legit' 'legit' 'dga' 'legit' 'legit' 'legit' 'dga' 'dga'
 'legit' 'legit' 'legit' 'legit' 'legit' 'legit' 'legit' 'dga' 'legit'
 'dga' 'legit' 'legit' 'dga' 'dga' 'legit' 'legit' 'legit' 'legit' 'legit'
 'dga' 'legit' 'dga' 'legit' 'dga' 'legit' 'legit' 'legit' 'dga' 'legit'
 'legit' 'dga' 'legit' 'dga' 'dga' 'legit' 'legit' 'dga' 'legit' 'legit'
 'legit']
[[1.00000000e+01 3.53116900e-01 5.31250096e+01 1.79876316e+00]
 [1.60000000e+01 9.90471212e-01 4.30964220e+01 1.10818477e+00]
 [1.60000000e+01 8.28927094e-01 2.92633115e+01 1.64302131e+00]
 [1.60000000e+01 8.28927094e-01 6.14159258e+01 1.21220419e+00]
 [1.40000000e+01 2.27128863e-01 9.11713948e+01 1.55723301e+00]
 [1.60000000e+01 9.90471212e-01 5.18879385e+01 1.12001541e+00]
 [2.00000000e+01 9.67922957e-01 6.03405378e+01 4.79308893e-01]
 [1.50000000e+01 4.87624030e-01 8.96848653e+01 6.09902031e-01]
 [1.50000000e+01 1.07095312e-01 9.98296262e+01 9.78998868e-01]
 [1.30000000e+01 3.53116900e-01 7.69730710e+01 1.91277276e+00]
 [1.80000000e+01 8.28927094e-01 2.63309751e+01 1.50002388e+00]
 [1.60000000e+01 8.28927094e-01 3.62796775e+01 1.32856827e+00]
 [1.90000000e+01 1.07095312e-01 1.52053365e+02 1.24778024e+00]
 [1.80000000e+01 1.07095312e-01 1.24480608e+02 9.87847090e-01]
 [1.10000000e+01 2.00584157e-03 3.60974662e+01 2.31347459e+00]
 [1.10000000e+01 8.28927094e-01 4.57005013e+01 1.54490609e+00]
 [1.60000000e+01 2.27128863e-01 8.87070193e+01 1.36624296e+00]
 [2.60000000e+01 3.67645424e-01 1.49729663e+02 7.35885365e-01]
 [1.00000000e+01 3.53116900e-01 3.77528108e+01 1.79860001e+00]
 [9.00000000e+00 1.48104030e-01 6.94372630e+01 1.32848330e+00]
 [1.20000000e+01 8.28927094e-01 5.91982208e+01 1.74767635e+00]
 [2.70000000e+01 6.67704770e-01 1.75709211e+02 6.87937468e-01]
 [3.10000000e+01 3.67645424e-01 1.74593075e+02 9.03314530e-01]
 [6.00000000e+00 1.48104030e-01 6.16895835e+00 2.80088869e+00]
 [1.80000000e+01 8.28927094e-01 2.10654440e+01 1.18073735e+00]]
['legit' 'dga' 'dga' 'dga' 'legit' 'dga' 'dga' 'legit' 'legit' 'legit'
 'dga' 'dga' 'legit' 'legit' 'legit' 'dga' 'legit' 'legit' 'legit' 'legit'
 'legit' 'dga' 'legit' 'legit' 'dga']
```

5.4 Đào tạo với các mô hình máy học

Ta sẽ sử dụng mô hình RandomForestClassifier để huấn luyện trên dữ liệu đã được chuẩn bị:

```
#5 RandomForestClassifier
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    3.4s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    3.8s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    4.2s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    4.8s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    5.2s finished
```

Và sử dụng mô hình VotingClassifier để kết hợp các mô hình đã được huấn luyện trước đó:

```
#8 VotingClassifier
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.2s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.5s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.9s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    1.5s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    1.9s finished
```

Các mô hình sử dụng bao gồm:

- LogisticRegression
- RandomForestClassifier
- GaussianNB
- ExtraTreesClassifier
- VotingClassifier

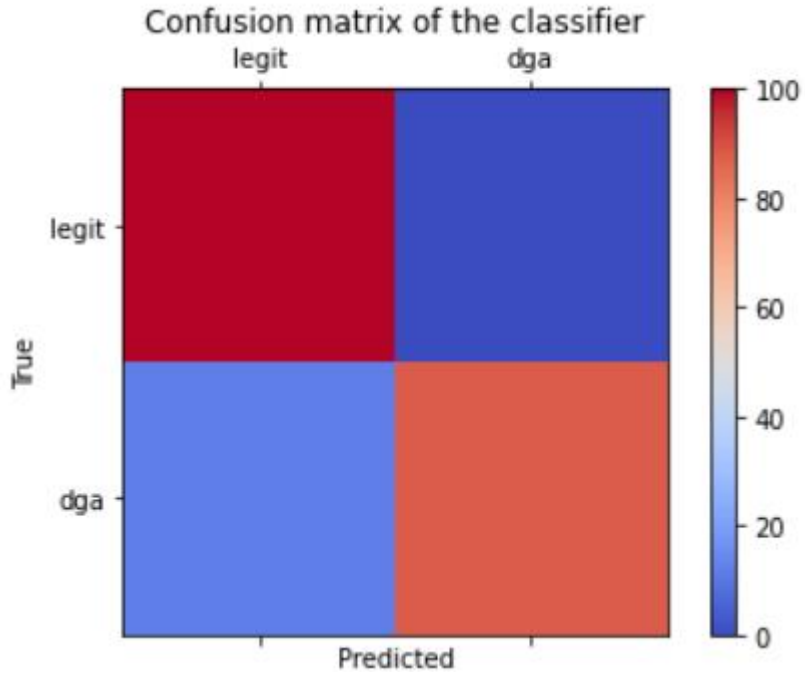
```
print('#4 LogisticRegression')

clf1 = LogisticRegression(random_state=1)
clf1.fit(X_train, y_train)


print('#5 RandomForestClassifier')

clf2 = RandomForestClassifier(bootstrap=True, max_depth=None, min_samples_leaf=1,
                              n_estimators=1500, n_jobs=40, oob_score=False,
                              random_state=1, verbose=1)
clf2.fit(X_train, y_train)

print('#6 GaussianNB')

clf3 = GaussianNB()
clf3.fit(X_train, y_train)

print('#7 ExtraTreesClassifier')

clf4 = ExtraTreesClassifier()
clf4.fit(X_train, y_train)

print('#8 VotingClassifier')

clf5 = VotingClassifier(estimators=[('lr', clf1), ('rf', clf2), ('gnb', clf3), ('etr', clf4)], voting='soft')
clf5.fit(X_train, y_train)
```

5.5 Thực hiện so sánh hiệu suất giữa các mô hình đã huấn luyện sử dụng các phương pháp cross-validation và hiển thị ma trận confusion cho mỗi mô hình:

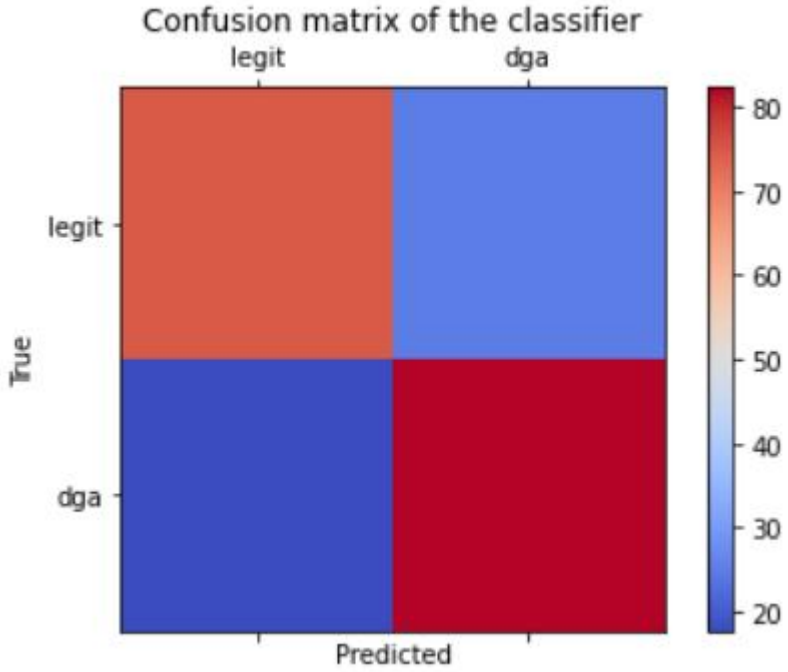
Quá trình so sánh:

```
#11 Comparison
Accuracy: 0.920000 (+/- 0.10) [Logistic Regression]
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.2s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.5s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.9s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    1.4s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    1.9s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.2s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.2s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.5s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    1.0s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    1.5s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    1.9s finished
```

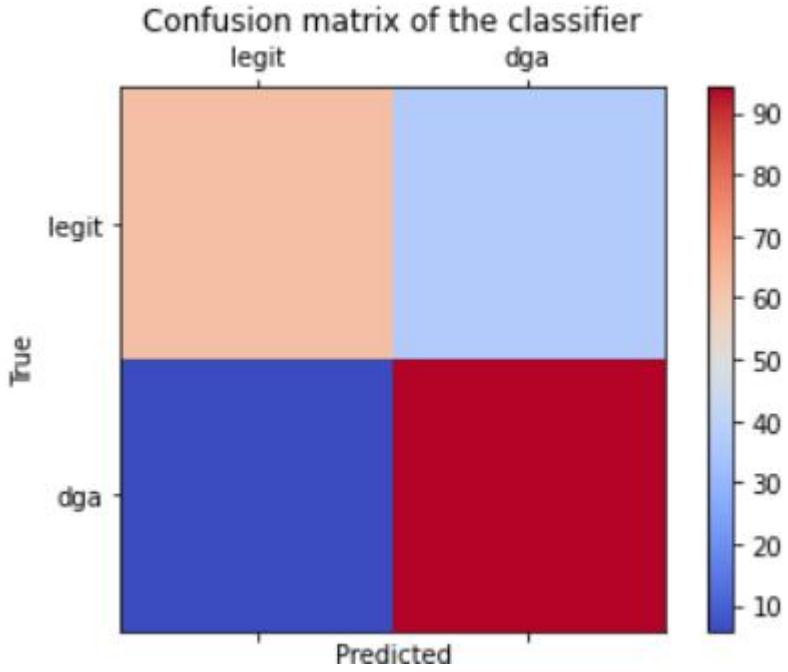
Kết quả các ma trận confusion tương ứng với từng mô hình:



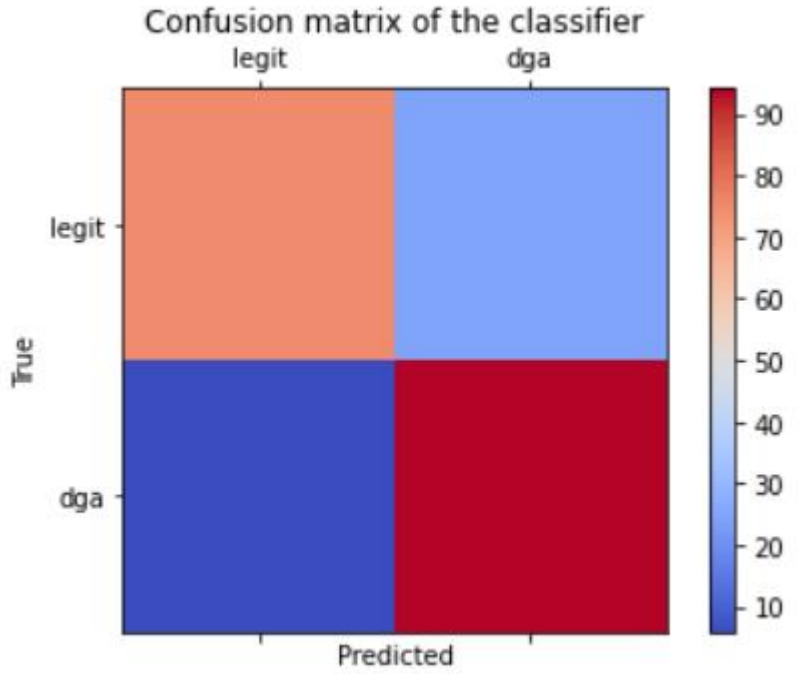
Confusion Matrix Stats
legit/legit: 100.00% (8/8)
legit/dga: 0.00% (0/8)
dga/legit: 11.76% (2/17)
dga/dga: 88.24% (15/17)



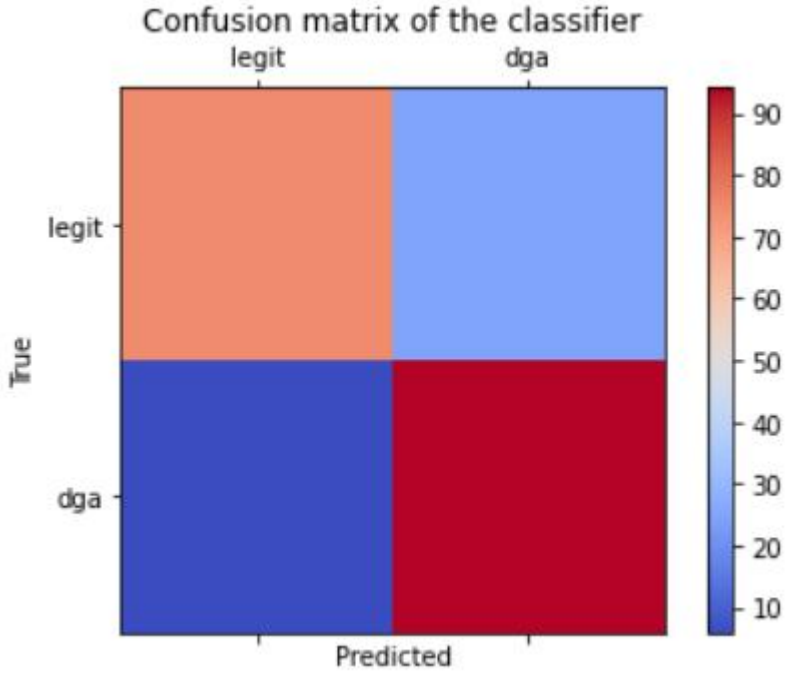
Confusion Matrix Stats
legit/legit: 75.00% (6/8)
legit/dga: 25.00% (2/8)
dga/legit: 17.65% (3/17)
dga/dga: 82.35% (14/17)



Confusion Matrix Stats
legit/legit: 62.50% (5/8)
legit/dga: 37.50% (3/8)
dga/legit: 5.88% (1/17)
dga/dga: 94.12% (16/17)



Confusion Matrix Stats
legit/legit: 75.00% (6/8)
legit/dga: 25.00% (2/8)
dga/legit: 5.88% (1/17)
dga/dga: 94.12% (16/17)



Confusion Matrix Stats
legit/legit: 75.00% (6/8)
legit/dga: 25.00% (2/8)
dga/legit: 5.88% (1/17)
dga/dga: 94.12% (16/17)

5.6 LIME, model-agnostic, local explainer

Bước tiếp sử dụng thư viện LIME (Local Interpretable Model-agnostic Explanations) để giải thích dự đoán của mô hình RandomForestClassifier cho một mẫu ngẫu nhiên trong tập kiểm thử:

Với 2 tên miền amazon.com và theirtheandaloneinto.com, kết qua được như hình bên dưới :

amazon.com là tên miền hợp pháp và mô hình đã dự đoán đúng.

theirtheandaloneinto.com là tên miền dga và mô hình đã dự đoán đúng là dga.

```
#12 LIME, model-agnostic, local explainer
DOMAIN = amazon.com
CharLength, TreeNewFeature, nGramReputation_Alexa, M
[10.          0.14810403 74.93908548  2.18440286]
Ground Truth ANSWER = legit
[Parallel(n_jobs=40)]: Using backend ThreadingBacker
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapse
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elaps
PREDICTION = legit
CORRECT Prediction :)
[Parallel(n_jobs=40)]: Using backend ThreadingBacker
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapse
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elaps
<IPython.core.display.HTML object>
[('CharLength ⩽ 12.00', 0.16217205131941997), ('71.
 ('0.11 < TreeNewFeature ⩽ 0.23', 0.045371364663055
{1: [(0, 0.16217205131941997), (2, 0.152966607801474
DOMAIN = theirtheandaloneinto.com
CharLength, TreeNewFeature, nGramReputation_Alexa, M
[ 24.          0.48762403 151.06379688   0.4815405
Ground Truth ANSWER = dga
[Parallel(n_jobs=40)]: Using backend ThreadingBacker
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapse
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elaps
PREDICTION = dga
CORRECT Prediction :)
```

Ví dụ khi mô hình dự đoán sai:

Với tên miền cũ là theirtheandaloneinto.com đầy là tên miền dga, nhưng mô hình dự đoán là legit, kết quả sẽ hiển thị:

WRONG Prediction :(

```
DOMAIN = theirtheandaloneinto.com
CharLength, TreeNewFeature, nGramReputation_Alexa, MinREBotnets
[ 24.          0.48762403 151.06379688   0.4815405 ]
Ground Truth ANSWER = dga

[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.1s finished

PREDICTION = legit
WRONG Prediction :(
```

5.7 SHAPE, Model agnostic with KernelExplainer

Sử dụng SHAP (SHapley Additive exPlanations) để giải thích dự đoán của mô hình RandomForestClassifier.

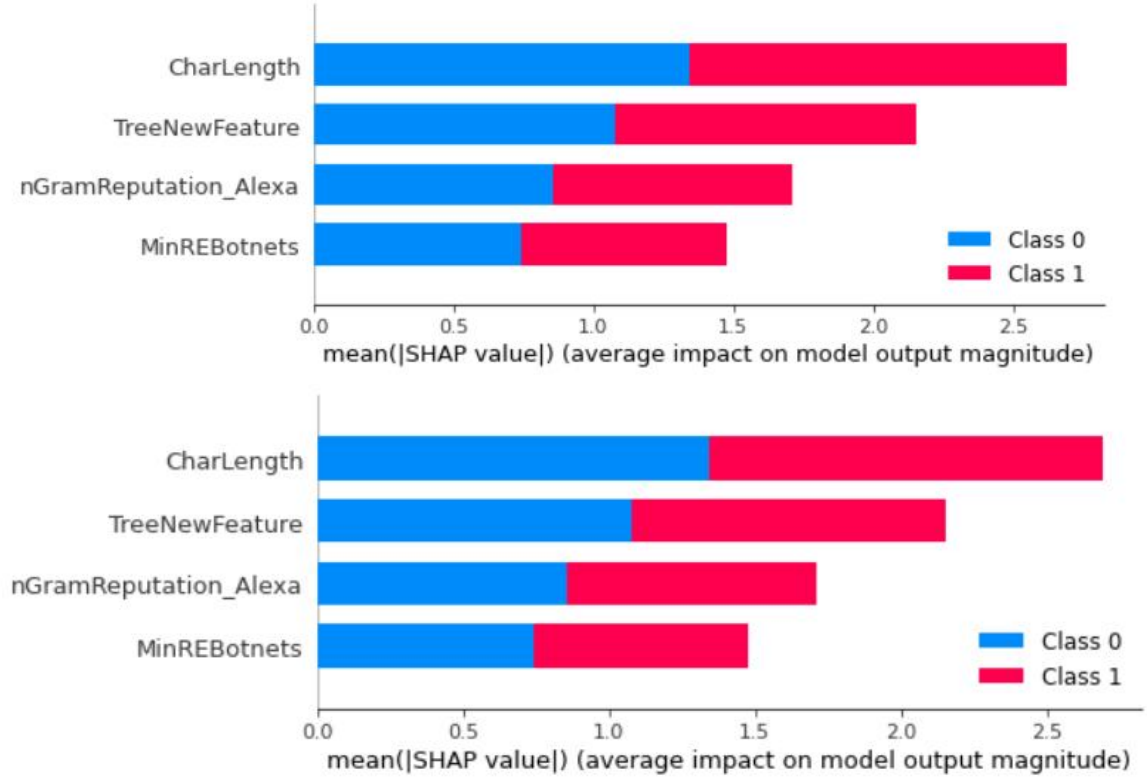
Quá trình thực hiện:

```
#13 SHAPE, Model agnostic with KernelExplainer
<IPython.core.display.HTML object>
    CharLength  TreeNewFeature  nGramReputation_Alexa  MinREBotnets
0         12.0        0.107095              69.870966      1.114496
1         20.0        0.367645             146.486376      0.939167
2         31.0        0.990471              24.725951      0.978133
3         19.0        0.107095             121.238159      0.977278
4          8.0        0.148104              23.282765      2.080281
..         ...             ...                    ...           ...
95        12.0        0.002006              53.183790      2.122443
96        13.0        0.828927              29.557642      1.636751
97         7.0        0.148104              29.250614      2.222434
98        18.0        0.107095             117.690077      1.114131
99        13.0        0.227129              63.438721      1.550019

[100 rows x 4 columns]
[[9.00000000e+00 1.48104030e-01 6.55184801e+01 2.12007503e+00]
 [1.20000000e+01 2.27128863e-01 5.94130050e+01 1.64152863e+00]
 [1.50000000e+01 1.07095312e-01 1.10480010e+02 1.22382741e+00]
 [2.00000000e+01 3.53116900e-01 3.87717916e+01 1.05357031e+00]
 [9.00000000e+00 1.48104030e-01 5.67742853e+01 2.23638311e+00]
 [3.00000000e+01 9.90471212e-01 3.54924701e+01 1.17010988e+00]
 [1.30000000e+01 3.53116900e-01 9.54470725e+01 1.73517732e+00]
 [1.00000000e+01 3.53116900e-01 2.66132927e+01 1.93098706e+00]
 [2.00000000e+01 9.90471212e-01 6.58825443e+01 8.22292190e-01]
 [1.10000000e+01 1.48104030e-01 7.00556712e+01 1.93992629e+00]
 [1.10000000e+01 1.07095312e-01 3.71116823e+01 1.21789230e+00]
```

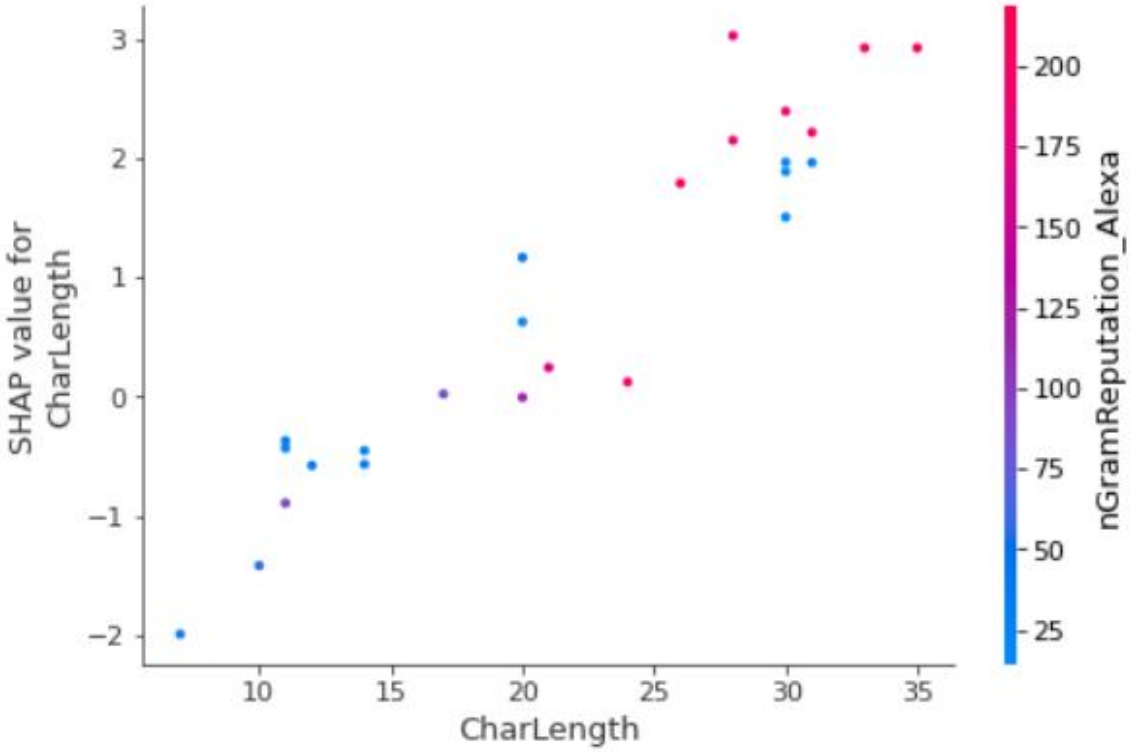
```
RandomForestClassifier(n_estimators=1500, n_jobs=40, random_state=1, verbose=1)
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.2s finished
  0%|                                                                                                  | 0/25 [00:00<?, ?it/s]
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.2s finished
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.2s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.3s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    0.4s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.5s finished
  4%|▍                                                                                                 | 1/25 [00:00<00:16,  1.48it/s]
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.0s
```

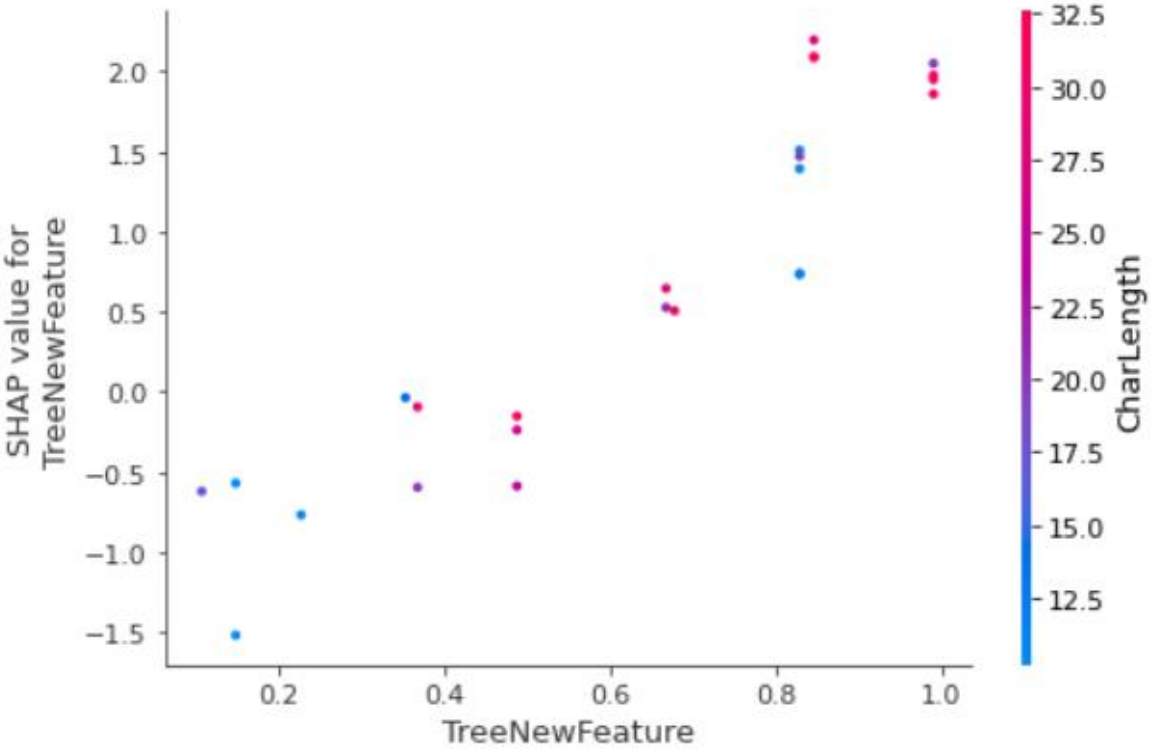
Kết quả:

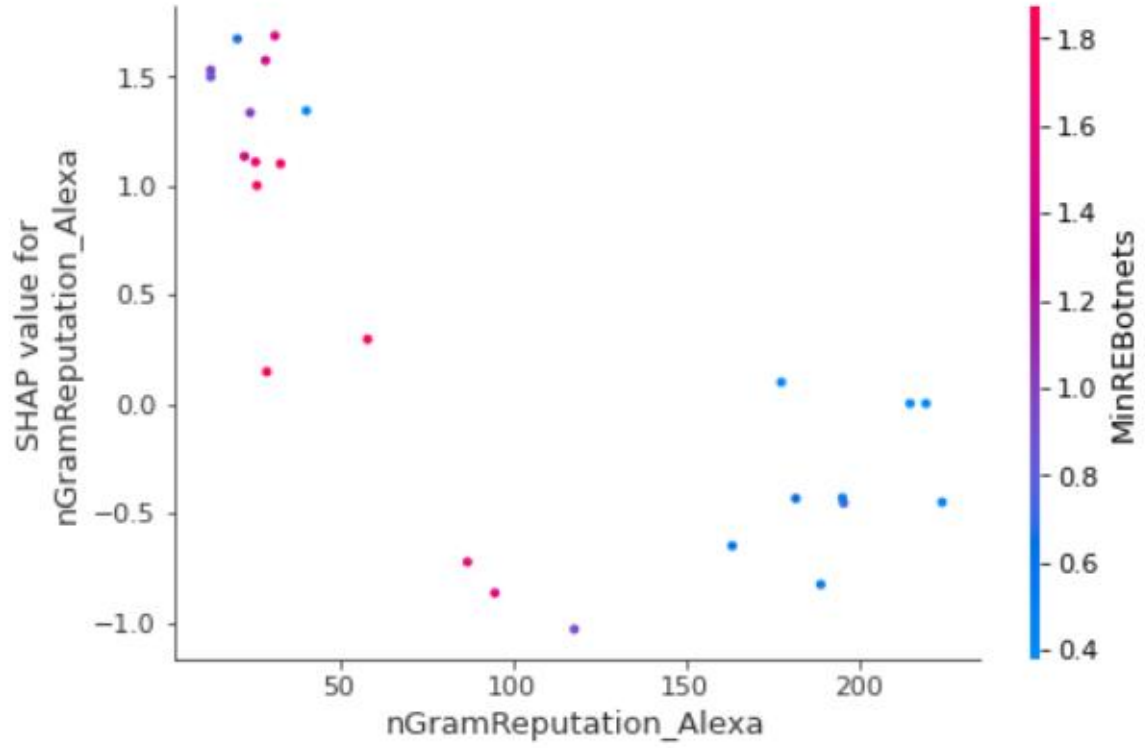
Hàm summarize plot:

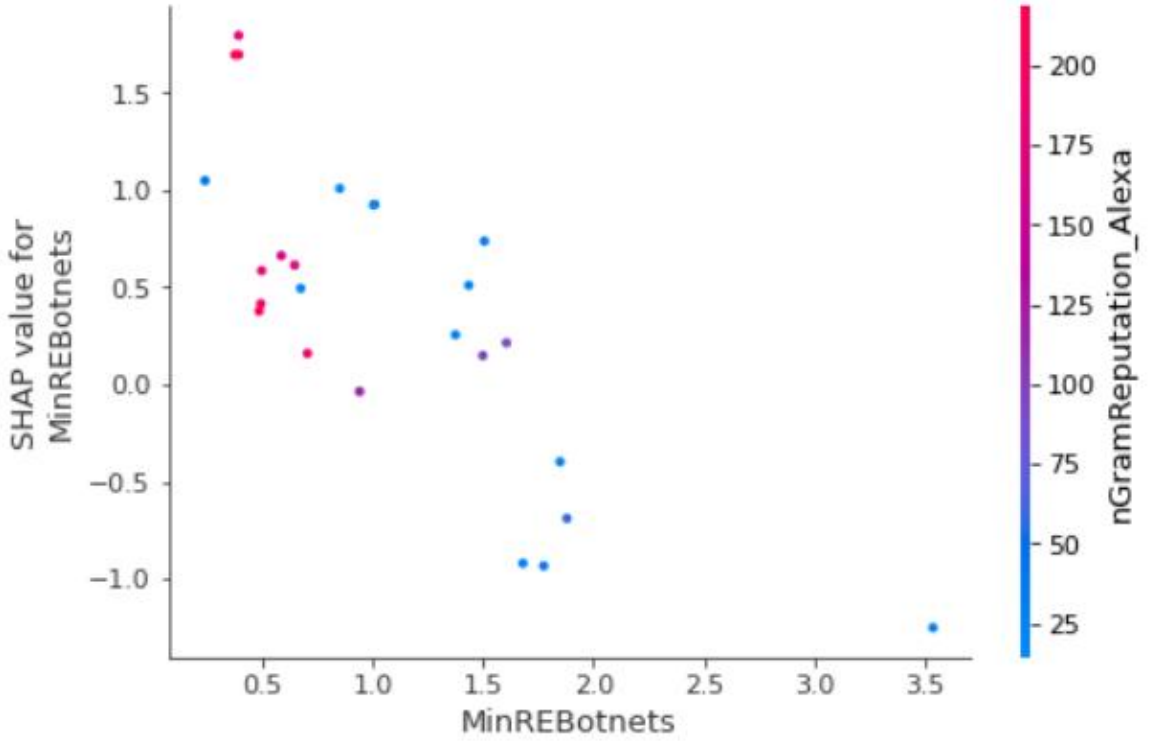


Hàm dependence plot:


SHAP value for CharLength
CharLength
nGramReputation_Alexa
3
2
1
0
−1
−2
10
15
20
25
30
35
200
175
150
125
100
75
50
25


SHAP value for TreeNewFeature
TreeNewFeature
CharLength
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
0.2
0.4
0.6
0.8
1.0
32.5
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5





5.8 What If & Counterfactual Explanations

Chọn ra một tập con nhỏ của dữ liệu từ data để sử dụng trong việc đào tạo và kiểm tra mô hình.

Những lời giải thích phản thực tế này mô tả sự thay đổi nhỏ nhất đối với thế giới có thể được thực hiện để đạt được kết quả mong muốn hoặc để đạt đến thế giới gần nhất có thể mà không cần giải thích logic bên trong của hệ thống.
Kết quả:

```
#14 What If & Counterfactual Explanations
                        Domainname  CharLength  TreeNewFeature  nGramReputation_Alexa  MinREBotnets  Label  LabelBinary
695670            hot4hairy2.tumblr.com          21        0.036657             118.100438      1.536609  legit            0
1063039                 uykbywcdoq.org          14        0.828927              20.413185      1.400194    dga            1
115721             americanhumanist.org          20        0.107095             140.040990      0.730273  legit            0
1399230                    jichoqok.ru          11        0.828927              29.608253      1.614117    dga            1
1549272                dedghicdykkh.com          16        0.828927              45.862521      1.465398    dga            1
1035787                   pugwmehj.org          12        0.828927              25.637575      1.494153    dga            1
429391                  chamada.com.br          14        0.353117             115.602766      1.821646  legit            0
3302                     webgains.com          12        0.107095              81.322099      1.422196  legit            0
534399                person-agency.ru          16        0.107095              93.508803      1.198789  legit            0
586743                    say-yess.com          12        0.148104              64.100884      2.272692  legit            0
288324             autosencuotas.com.ar          20        0.107095             140.562801      0.947172  legit            0
911193           prosoundandvision.co.uk          23        0.107095             177.640271      0.940750  legit            0
1360105                    rixihilk.ru          11        0.353117              30.355694      1.893885    dga            1
1034471                  oxztytzs.com          12        0.353117              37.046225      2.089359    dga            1
316617                    swarzedz.pl          11        0.828927              30.414841      1.847376  legit            0
839675                    777coin.com          11        0.148104              61.684743      2.974476  legit            0
1413081        withlightlawspublish.com          24        0.107095             143.944594      1.218152    dga            1
924592                         zic9.kr           7        0.148104               5.918785      3.148297  legit            0
1632023  campheatinvitefightsucceed.com          30        0.846071             172.343216      0.515703    dga            1
136711                     aellune.com          11        0.148104              67.148358      1.591378  legit            0
1767523             eikkyaasooegyioy.org          20        0.227129              52.181357      0.764879    dga            1
957630                       asrd.org           8        0.148104              34.821387      1.910144  legit            0
356702                 chichester.ac.uk          16        0.107095             108.337635      1.051413  legit            0
30232                 tarafsizhaber.com          17        0.107095             100.784549      1.212907  legit            0
1174643              hxkhctbcshyxuh.biz          18        0.828927              24.468398      1.542751    dga            1
695670    0
1063039   1
115721    0
1399230   1
1549272   1
1035787   1
429391    0
3302      0
534399    0
586743    0
288324    0
911193    0
1360105   1
1034471   1
```

5.9 Anchor explanations

Ta sử dụng thư viện alibi để tạo giải thích dự đoán của mô hình thông qua phương pháp Anchor.

```
#14 Anchor explanations
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    0.2s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.2s finished
[[2.80000000e+01 9.90471212e-01 3.89758841e+01 9.77627303e-01]
 [1.20000000e+01 1.07095312e-01 5.58710304e+01 1.06866762e+00]
 [1.10000000e+01 3.53116900e-01 2.93217418e+01 1.83119661e+00]
 [1.40000000e+01 1.07095312e-01 1.08714181e+02 1.00681893e+00]
 [2.00000000e+01 6.67704770e-01 6.38810438e+01 6.50597001e-01]
 [2.00000000e+01 6.67704770e-01 1.56699688e+02 5.87072307e-01]
 [2.00000000e+01 9.67922957e-01 4.51729663e+01 5.25407419e-01]
 [2.50000000e+01 3.67645424e-01 1.51481809e+02 8.75893292e-01]
 [2.40000000e+01 3.67645424e-01 1.54193082e+02 8.16406252e-01]
 [9.00000000e+00 3.53116900e-01 1.64411059e+01 1.95766964e+00]
 [1.70000000e+01 9.90471212e-01 2.43458765e+01 9.55364070e-01]
 [1.00000000e+01 3.53116900e-01 4.93010096e+01 1.73453743e+00]
 [2.00000000e+01 8.28927094e-01 2.55267926e+01 7.76181014e-01]
 [1.10000000e+01 1.48104030e-01 3.38731155e+01 2.26139410e+00]
 [1.10000000e+01 1.07095312e-01 5.64982792e+01 8.02035366e-01]
 [3.00000000e+01 9.90471212e-01 2.70355515e+01 9.56489979e-01]
 [2.40000000e+01 1.07095312e-01 1.91396707e+02 8.05671011e-01]
 [1.90000000e+01 1.07095312e-01 1.48980710e+02 1.18727251e+00]
 [1.60000000e+01 1.07095312e-01 1.22663645e+02 1.28774991e+00]
 [9.00000000e+00 3.53116900e-01 4.06798610e+01 2.39255432e+00]
 [1.80000000e+01 8.28927094e-01 3.55030328e+01 1.19920436e+00]
 [1.70000000e+01 1.07095312e-01 1.24072040e+02 1.15474537e+00]
 [2.20000000e+01 1.07095312e-01 1.63242848e+02 7.74732659e-01]
 [1.70000000e+01 1.07095312e-01 1.38732225e+02 1.53099414e+00]
 [3.20000000e+01 6.67704770e-01 2.07672112e+02 6.39685901e-01]
 [1.40000000e+01 1.07095312e-01 9.56950759e+01 1.12788322e+00]
```

Giải thích với ANCHOR:

Với tên miền autosencuotas.com.ar là tên miền legit và mô hình đã dự đoán đúng là
legit.

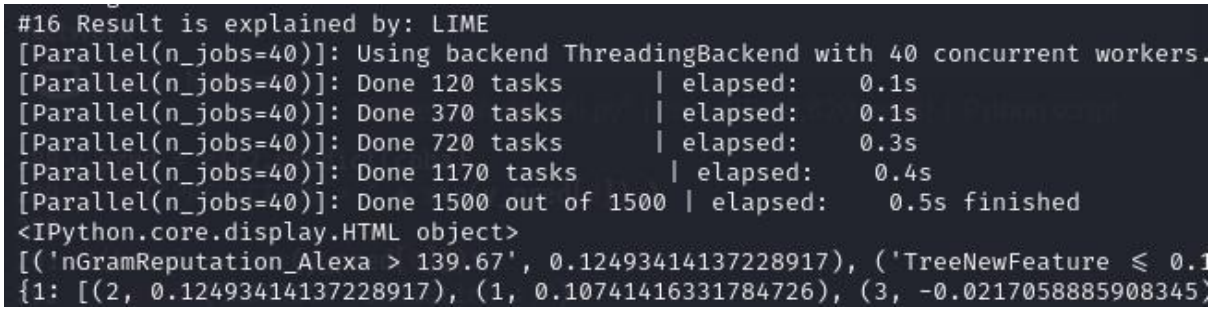
```
autosencuotas.com.ar
Ground Truth = legit
[2.00000000e+01 1.07095312e-01 1.40562801e+02 9.47171889e-01]
[Parallel(n_jobs=40)]: Using backend ThreadingBackend with 40 concurrent workers.
[Parallel(n_jobs=40)]: Done 120 tasks      | elapsed:    0.0s
[Parallel(n_jobs=40)]: Done 370 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.2s finished
Prediction:  legit
```

5.9 Kết quả đánh giá của các XAI về mô hình

ANCHOR:

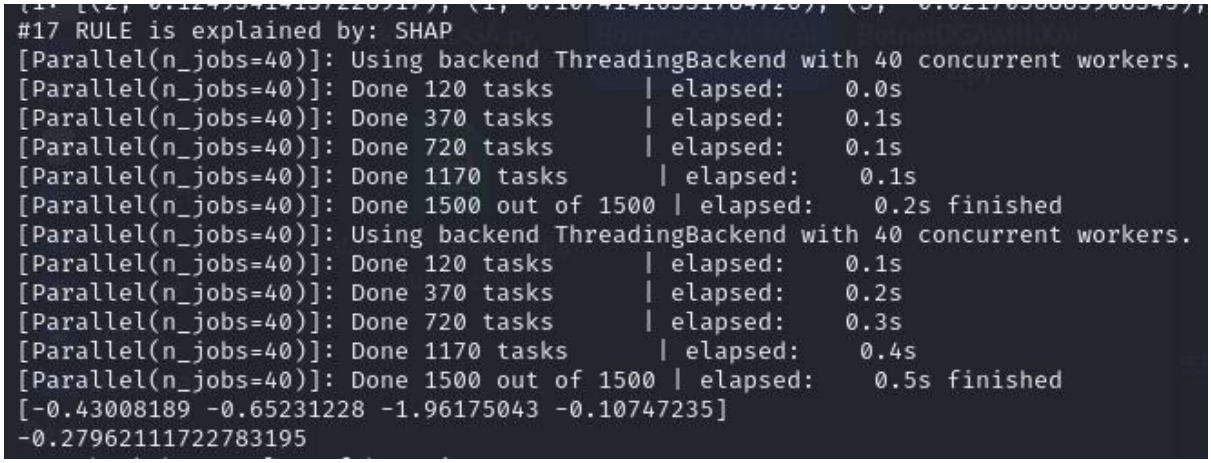
```
[Parallel(n_jobs=40)]: Done 720 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1170 tasks      | elapsed:    0.1s
[Parallel(n_jobs=40)]: Done 1500 out of 1500 | elapsed:    0.2s finished
Anchor: nGramReputation_Alexa > 88.87 AND CharLength ⩽ 23.25
Precision: 0.99
Coverage: 0.30
```

LIME:



<IPython.core.display.HTML object>
[('nGramReputation_Alexa > 139.67', 0.12493414137228917), ('TreeNewFeature <= 0.11', 0.10741416331784726), ('0.86 < MinREBotnets <= 1.09', -0.0217058885908345), ('16.50 < CharLength <= 21.00', 0.016747567518053445)]
{1: [(2, 0.12493414137228917), (1, 0.10741416331784726), (3, -0.0217058885908345), (0, 0.016747567518053445)]}

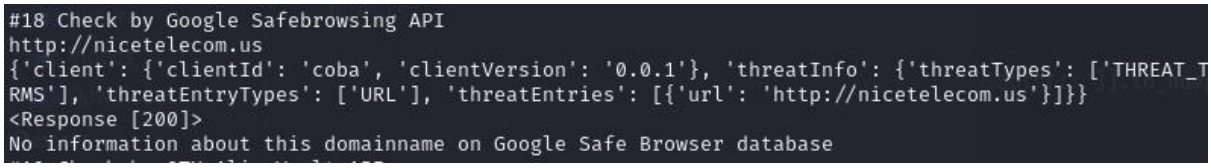
SHAPE:



5.10 OSINT: Google Safebrowsing API và OTX AlienVault API

Kiểm tra với Google Safebrowsing API:

Với tên miền nicetelecom.us, Kết quả trả về là 200 đây là tên miền legit.

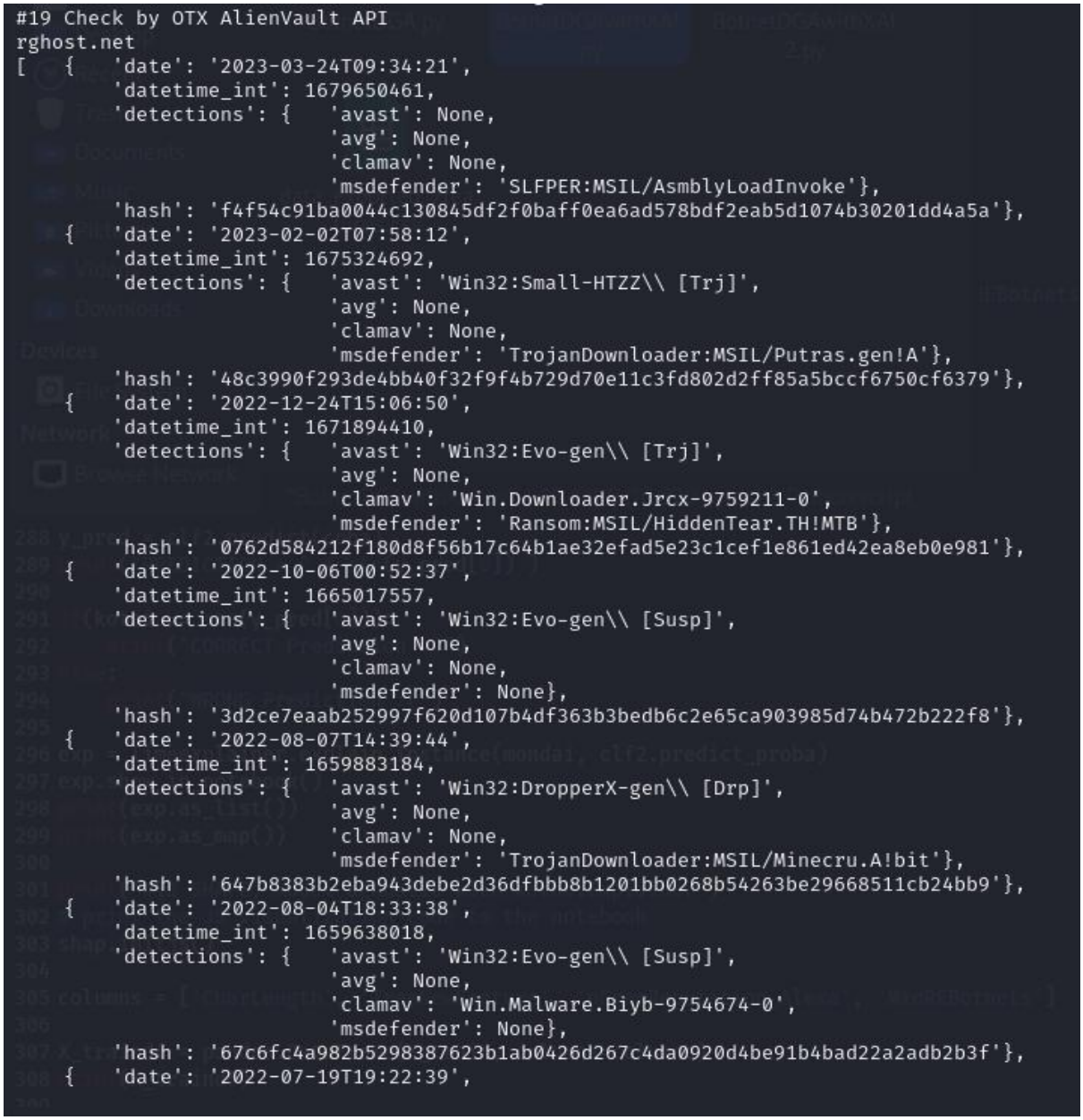


{'client': {'clientId': 'coba', 'clientVersion': '0.0.1'}, 'threatInfo': {'threatTypes': ['THREAT_TYPE_UNSPECIFIED', 'MALWARE', 'SOCIAL_ENGINEERING', 'UNWANTED_SOFTWARE', 'POTENTIALLY_HARMFUL_APPLICATION'],

'platformTypes': ['ALL_PLATFORMS'], 'threatEntryTypes': ['URL'], 'threatEntries': [{'url': 'http://nicetelecom.us'}]}}

Kiểm tra với OTX AlienVault API:

Với tên miền rghost.net, Kết quả kiểm tra thấy được tên miền có liên quan đến các mối đe dọa như trong hình, đây là tên miền dga.



CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Ở chương này, chúng tôi đưa ra những kết luận về nghiên cứu, những hạn chế, và đồng thời đưa ra hướng cải thiện và phát triển.

6.1. Kết luận

Chúng tôi giới thiệu một mô hình mới để phát hiện DGA botnet. Mô hình rừng ngẫu nhiên đạt độ chính xác 96,3% (được thử nghiệm với bộ dữ liệu của 55 họ DGA botnet) và vượt trội so với công việc trước đó. Mô hình này cũng mạnh mẽ hơn trước ba cuộc tấn công đối nghịch DGA tiên tiến (MaskDGA, CharBot và DeepDGA) so với các mô hình trước đó.

Nghiên cứu đã nhấn mạnh tính thực tiễn của việc kết hợp XAI và OSINT để mang lại khả năng giải thích AI tốt hơn thông qua các phương pháp tiếp cận ý kiến thứ hai, từ đó bắt chước hiện tượng ý kiến thứ hai trong các tình huống bệnh viện/y tế để xác nhận kết quả/phát hiện. Sự kết hợp XAI và OSINT như phương pháp giải quyết cho sự hoài nghi đối với đầu ra của mô hình, điều này có thể góp phần tạo nên sự tin cậy của hệ thống CTI và ngăn chặn sự thiên vị tự động hóa khi người dùng tin tưởng quá nhiều vào đầu ra của hệ thống CTI. Việc kết hợp XAI và OSINT cũng có khả năng giải quyết các vấn đề về cờ giả.

Việc kết hợp XAI và OSINT để cải thiện niềm tin có thể dẫn đến hiện tượng chuyển đổi mô hình. Các cộng đồng an ninh mạng sẽ rời khỏi mô hình chia sẻ CTI truyền thống (chỉ chia sẻ các chỉ báo về mối đe dọa, chẳng hạn như tên miền của mối đe dọa) và các cộng đồng sẽ bắt đầu chia sẻ mô hình AI/ML cho các hệ thống CTI. Với sự xuất hiện của mô hình chia sẻ CTI có thể tính toán, sự hợp tác bổ sung giữa các cộng đồng an ninh mạng sẽ diễn ra để phát triển các hệ thống CTI dựa trên AI/ML tiên tiến. Ví dụ: sử dụng các kỹ thuật học chuyển giao để phát triển AI/ML mới cho các nhiệm vụ/vấn đề an ninh mạng mới bằng cách sử dụng các mô hình được chia sẻ.

6.2. Hướng phát triển

Bên cạnh những kết quả mà chúng tôi đã đạt được trong nghiên cứu này thì vẫn còn nhiều vấn đề có thể tiếp tục cải tiến, phát triển trong tương lai. Hạn chế của mô hình phát hiện DGA của chúng tôi là độ phức tạp về thời gian khi tính toán các đặc trưng và sự hạn chế trong việc đối phó với các cuộc tấn công MaskDGA.

Cải tiến trong tương lai nên tập trung vào việc tạo ra các đặc trưng tốt hơn và có một chiến lược phòng thủ đối nghịch. Phòng thủ mục tiêu di động (MTD) có khả năng nâng cao độ tin cậy của mô hình bằng cách kết hợp nhiều mô hình khác nhau để hoạt động cùng nhau.

TÀI LIỆU THAM KHẢO

H. Suryotrisongko, Y. Musashi, A. Tsuneda and K. Sugitani, "Robust Botnet DGA Detection: Blending XAI and OSINT for Cyber Threat Intelligence Sharing," in IEEE Access, vol. 10, pp. 34613-34624, 2022, doi: 10.1109/ACCESS.2022.3162588.